

Atte Väätäinen

# RESPONSIIVINEN VERKKOSUUNNITTELU

Opinnäytetyö  
Tietojenkäsittely


Tammikuu 2013




**MIKKELIN AMMATTIKORKEAKOULU**

Mikkeli University of Applied Sciences

## KUVAILULEHTI

 <p><b>MIKKELIN AMMATTIKORKEAKOULU</b> Mikkeli University of Applied Sciences</p>		<b>Opinnäytetyön päivämäärä</b>  22.02.2013	
<b>Tekijä(t)</b> Atte Väätäinen		<b>Koulutusohjelma ja suuntautuminen</b> Tietojenkäsittelyn koulutusohjelma	
<b>Nimeke</b> Responsiivinen verkkosuunnittelu			
<b>Tiivistelmä</b>  <p>Tässä opinnäytetyössä tutkitaan responsiivisen verkkosuunnittelun mahdollisuuksia ja mobile first -ajattelumallin hyötyjä. Työn tavoitteena oli luoda kaksi erilaista verkkosivudemoa, jotka esittelevät HTML5:n ja CSS3:n ominaisuuksien lisäksi responsiivisen verkkosuunnittelun ajatusmallia.</p> <p>Verkkosivujen suunnittelijat voivat nykyään harvoin tietää, millä laitteella heidän julkaisuaan tullaan selaamaan. Kun se on julkaistu, käyttäjiä voi vierailta palvelussa kännyköillä, tableteilla, työpöytäkoneilla tai television ruudulta. Tämä näyttöpäätteiden kokoerojen laaja kirjo aiheuttaa verkkosuunnittelijoille uudenlaisia haasteita. Jokaiselle laitteelle oman palvelun suunnittelu ei myöskään usein ole taloudellisesti kannattavaa. Näiden ongelmien takia Ethan Marcotte on esitellyt uuden responsiivisen tavan suunnitella verkkopalveluita.</p> <p>Ethanin vuonna 2010 kirjoittaman kirjan Responsive web design - julkaisun jälkeen, suunnittelijat ovat vähitellen alkaneet ymmärtämään kyseisen työtavan hyödyt. Aihe on myös ajankohtainen siinä mielessä, että mobiililaitteiden selaimissa on yleisesti ottaen hyvä tuki CSS:lle, HTML:lle ja Javascriptille, joten luotuasi responsiiviset verkkosivut, ne eivät jää vain tietyn kehittyneen päätelaitteen hienoudeksi, vaan suurimmalla osalla esimerkiksi älypuhelimista on mahdollisuus niitä hyödyntää.</p>			
<b>Asiasanat (avainsanat)</b>  Web Design, HTML5, CSS3			
<b>Sivumäärä</b> 38	<b>Kieli</b> Suomi	<b>URN</b>	
<b>Huomautus (huomautukset liitteistä)</b>			
<b>Ohjaavan opettajan nimi</b> Jukka Selin		<b>Opinnäytetyön toimeksiantaja</b> Mikkelin ammattikorkeakoulu	

## DESCRIPTION

 <p><b>MIKKELIN AMMATTIKORKEAKOULU</b> Mikkeli University of Applied Sciences</p>		<b>Date of the bachelor's thesis</b>  22 February 2013	
<b>Author(s)</b>		<b>Degree programme and option</b> Business Information Technology	
<b>Name of the bachelor's thesis</b> Responsive Web design			
<b>Abstract</b>  <p>In this thesis I examined the abilities and benefits of responsive web design and using the concept of mobile-first. The goal of the thesis was to create two different types of web demos that demonstrate HTML5 and CSS3 features, and follows the idea of responsive web design.</p> <p>Web designers can nowadays rarely know with which device their publication will be browsed. After the publishing users can visit the service with mobile phones, tablets, desktop computers or television screen. This range of display device sizes create challenges to web designers. Also creating a specific service to every platform is not profitable. The solution to these problems Ethan Marcotte has presented a new responsive way of designing web services.</p> <p>After the release of Ethan's book Responsive web desing, written in 2010, designers have started to understand the benefits of responsive design. The topic is also current because mobile browsers have generally good support for CSS, HTML and JavaScript. After publishing the websites majority of smartphone owners have the opportunity to browse them.</p>			
<b>Subject headings, (keywords)</b>  Pattern, graphic design, collection, textile printing			
<b>Pages</b>	<b>Language</b> Finnish	<b>URN</b>	
<b>Remarks, notes on appendices</b>  responsive, web design, HTML5, CSS3			
<b>Tutor</b>		<b>Bachelor's thesis assigned by</b> -	

# SISÄLTÖ

1	JOHDANTO .....	1
2	VERKKOSIVUJEN SUUNNITTELU .....	2
2.1	Tyyli ja sisältö.....	2
2.2	Rakenteet .....	3
2.3	Osoitinlaitteet.....	5
3	RESPONSIIVINEN VERKKOSUUNNITTELU.....	6
3.1	One web ja mobile first.....	7
3.1.1	Mobiililaitteiden verkon mahdollisuudet.....	8
3.1.2	Natiivit sovellukset vs. mobiiliverkko sovellukset.....	9
3.2	CSS3 .....	10
3.3	HTML5 .....	13
3.4	Joustava sommittelu.....	15
3.5	Joustavat kuvat.....	17
4	CASE: PORTFOLIOT .....	18
4.1	Portfolioiden vaatimusmäärittely ja suunnittelu.....	19
4.2	Käytännön toteutus (portfolio1) .....	20
4.3	Käytännön toteutus (portfolio 2) .....	28
4.4	Portfolioiden testaus .....	34
5	PÄÄTÄNTÖ .....	34
	LÄHTEET .....	36

## 1 JOHDANTO

Internetin käyttöön kodin ja työn ulkopuolella sopivat laitteet ja nettiyhteydet yleistyvät. Näitä mobiileja päätelaitteita, eri kokoisine näyttöineen ja ominaisuuksineen, on tuhansia. Älypuhelin ja tablettien myynti on hurjassa kasvussa ja ihmiset selaavat niillä webiä mitä erilaisimmissa tilanteissa. Osa kotisohvalla makaillessa, ja toiset kiireessä työmatkalla.

Tämän päätelaitteiden kirjon takia on selvää, ettei verkkosuunnittelijan kannata suunnitella verkkosivuja jokaiselle päätelaitteelle erikseen. Jo pelkästään erillisen mobiilisivuston toteuttaminen tarkoittaa käytännössä kaksinkertaista työmäärää ja ylläpito-kustannuksia. Näihin haasteisiin vastaa responsiivinen suunnittelu, joka tarkoittaa käytännössä vain uutta joustavampaa tapaa suunnitella sivut. Tekniikat tämän toteuttamiseen ovat jo olleet olemassa pitkään.

Tämän työn tarkoitus oli osoittaa kahdella esimerkillä responsiivisen verkkosuunnittelun ja mobile first -ajattelutavan mahdollisuuksia ja esitellä myös HTML5:n ja CSS3:n uusia ominaisuuksia. Työ sisältää näiden *demojen* ulkoasun suunnittelun ja toteutuksen. Toteutuksessa on otettu huomioon mahdollisimman laaja selaintuki, ja vaikka HTML5:n ja CSS3:n uusia ominaisuuksia esitelläänkin, on esimerkkisivut pyritty rakentamaan niin, että ne toimivat käyttäjystävällisesti myös selaimilla, jotka eivät tue näitä uusia tekniikoita. Vaikka internet on pullollaan erilaisia valmiita sivustopohjia ja julkaisujärjestelmiä, tämän työn tarkoitus on ollut rakentaa esimerkkidemot alusta alkaen, ja rajauksena onkin ollut, että valmiita pohjia ei käytetä, ja jQueryn tapaisia kirjastojakin mahdollisimman vähän. Toisena rajauksena jätetään pois esteettömyyteen liittyvät asiat. Demot toteutetaan toki W3C,- sekä HTML-standardien mukaisesti, mutta esimerkiksi erillisiä testejä erilaisia rajoituksia omaaville käyttäjäryhmille ei suoriteta.

Työn tärkeimpänä pohjana on ollut Ethan Macrotten (2010) kirjoittama kirja *Responsive web design*, sekä Luke Wroblewskin (2011) *Mobile first*.

## 2 VERKKOSIVUJEN SUUNNITTELU

Verkkosivujen suunnittelulla tarkoitetaan koko verkkosivujen luontiprosessia. Prosessi pitää sisällään ainakin visuaalisen ilmeen- ja sisällön suunnittelun, sekä kehitystyön. Verkkosivuja suunniteltaessa on ensisijaisen tärkeää lähteä käyttäjälähtöisestä suunnittelusta. Helppo navigointi, miellyttävä visuaalinen ilme, johdonmukaisuus, mielenkiintoinen sisältö ja selkeät rakenteet takaavat, että käyttäjä palaa sivulle uudelleen.

Käyttäjälähtöisessä suunnittelussa suunnittelijan tulisi miettiä mahdollisia käyttötapa-uksia, esimerkiksi jos käyttäjä on tullut sivuille ensimmäistä kertaa Googlen hakutuloksen seurauksena, hänelle pitäisi nopeasti selvittää, mitä tuotteita tai palveluita sivustosi tarjoaa, muuten potentiaalinen asiakas etsii seuraavan Google-linkin. Kuluttajan tarpeisiin keskittynyt sivusto saa usein käyttäjät myös palaamaan uudelleen sivuille.

Liian tiukkojen oletusten tekeminen käyttäjistä saattaa kuitenkin vaikuttaa sivun esteettömyyteen, joten sitä kannattaa pyrkiä välttelemään (Mitä on verkkosuunnittelu? 2010).

### 2.1 Tyyli ja sisältö

Verkkosuunnittelijan täytyy aina tasapainotella tyylin, sisällön, ja informaation välillä. Liiallinen informaation määrä ja täyteen pakatut sivut ovat yleensä varmoja tapoja käännyttää käyttäjät pois verkkosivuilta. Onkin todettu, että käyttäjän täytyisi pystyä arvioimaan sivuston käyttötarkoitus alle neljässä sekunnissa, muuten informaatiota on liikaa tai visuaalinen ilme on sekava (Kahdeksan yleistä virhettä 2010).

Verkkosuunnittelussa tyyli ja sisältö erotetaan myös konkreettisesti toisistaan. Tämä tapahtuu yleensä luomalla CSS -tyylisivuja, joissa määritellään, miltä sivuston tulisi näyttää käyttäjälle. Tämä mahdollistaa saman informaation tarjoamisen usealle eri päätelaitteelle sopivassa muodossa, vain tyyli tiedostoja muokkaamalla.

Visuaalisesti ja sisällöllisesti ajan tasalla olevilla verkkosivuilla voidaan viestittää yrityksen elinvoimaisuudesta. Onkin tärkeää, että jo suunnitteluvaiheessa otetaan

huomioon myös päivitys- sekä ylläpitomahdollisuudet. Usein aliarvioidaankin sivujen ajan tasalla pitämiseen kuluvaan aikaa ja muita resursseja. (Oikea lähestymistapa 2011)

Latausnopeuksien ja tiedostokoon optimointi hävisi joksikin aikaa verkkosuunnittelukeskustelusta nopeiden laajakaistayhteyksien tultua lähes joka kotiin. Nyt mobiili laitteiden ja niillä suoritettujen tiedonsiirron yleistyttyä on tämän kaltainen optimointi entistä tärkeämpää. Verkkosuunnittelijan täytyykin kysyä itseltään, onko mobiilikäyttäjän esimerkiksi järkevää ladata kaikki kuvat näennäisen saumattoman selaamisen saavuttamiseksi vai voisiko osan raskaasta datasta laittaa esimerkiksi linkkien taakse (Taali Marko 2010). Muun muassa Google onkin tehnyt aiheesta tutkimuksia, ja todennut että hitaat latausajat käännäyttävät mobiililaitteilla Internetiä selaavien lisäksi myös perinteisellä pöytäkoneella selaintaan tutkivat käyttäjät pois. (Using site speed in web search ranking 2010)

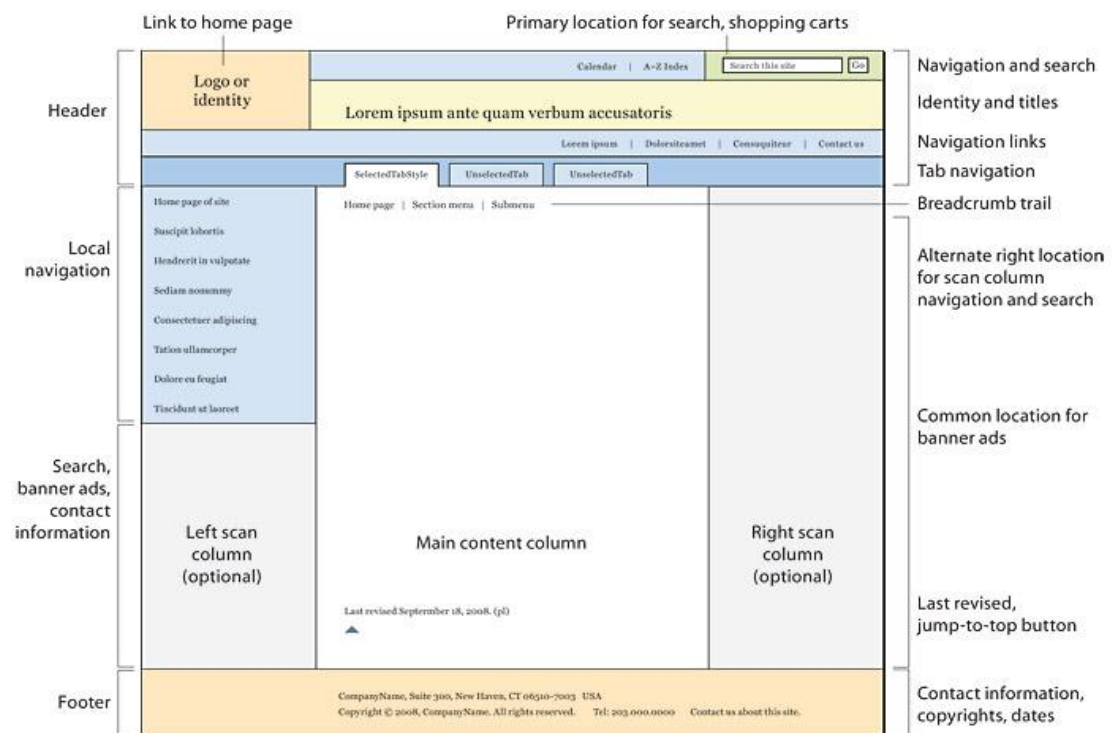
Verkkosuunnittelijoiden avuksi on ilmestynyt erilaisia avoimen lähdekoodin, sekä maksullisia julkaisu- ja sisällönhallintajärjestelmiä. Julkaisujärjestelmä on verkkosivustolle asennettava ohjelma, jonka avulla voidaan ylläpitää verkkosivujen sisältöä, sekä verkkokauppaa ja muokata valikoita ja asentaa uusia toimintoja. Julkaisujärjestelmissä on usein graafinen käyttöliittymä sisällön lisäämistä varten, ja se sisältää usein tekstinkäsittelyohjelmista tuttuja muokkaustoimintoja. Tämä auttaa henkilöitä, jotka eivät osaa välttämättä koodata, julkaisemaan ja ylläpitämään esimerkiksi blogeja tai muuten vain omia verkkosivujaan. Suosittuja julkaisujärjestelmiä ovat Joomla, Drupal ja Wordpress. Vaikka julkaisujärjestelmän käyttäminen onkin usein validi vaihtoehto, on ne rajattu tämän opinnäytetyön ulkopuolelle, sillä responsiivisen verkkosivun luonti on usein parempi aloittaa puhtaalta pöydältä. (Julkaisujärjestelmät 2012)

## **2.2 Rakenteet**

Verkkosivut sisältävät sisällöllisen ja visuaalisen rakenteen, ja nämä kaksi tulisi erottaa toisistaan selvästi. Sisältörakenteessa keskitytään sivuilla tarjottavaan tietoon, kun taas visuaalisessa rakenteessa keskitytään ulkoasuun ja käytettävyyteen. Verkkosivujen rakenteen tulisi olla yhtenäinen, kun käyttäjä hahmottaa etusivun rakenteen, niin hänen ei tarvitse taas seuraavalla sivulla käyttää ylimääräistä aikaa rakenteen hahmottamiseen.

Hyvä rakenne merkitään web-standardeja käyttäen niin, että sivuston sisältö kuvataan HTML - tai XHTML -tiedostoissa, jossa määritellään sivun tekstit, metatiedot, videot, kuvat ja muu sisältö. Tässä tiedossa määritellään myös, mistä sivun käyttämä ulkoasu haetaan, eli yleensä CSS lähde. (Larvanko 2007)

Sivuston hyvä rakenne vaikuttaa käytön miellyttävyyden lisäksi myös saavutettavuuteen, nopeampiin latausaikoihin ja hakukoneet löytävät sivut paremmin. Jotkut verkkosuunnittelijoiden (ja käyttäjien) hyväksi havaitsemat rakenteet ovatkin jo yleistyneet lähes standardeiksi. Lähes aina sivuilta löytyy ainakin *headeri*, *navigaatio*, etsi-toiminto, *footeri* sekä sisältöalue (Page structure and site design 2009). Hyvä verkkosivujen rakenne parantaa myös esteettömyyttä. Esteettömyydellä tarkoitetaan käyttäjien ja toimintatilanteiden monimuotoisuuden huomioon ottamista (ESSI). Esteettömät verkkosivut saavutetaan noudattamalla W3C sisällöntuotannon ja HTML:n standardeja, erottelemalla sisältö ja tyylit, sekä testaamalla sivustoja erilaisia rajoituksia omaavilla käyttäjäryhmillä. Kuvassa 1 nähdään tyypillinen verkkosivun rakenne.



**KUVA 1. Tyypillinen verkkosivun rakenne (Page structure and site design 2009)**



### 2.3 Osoitinlaitteet

Tietokoneita käytetään useimmiten graafisilla käyttöliittymillä, joissa objektit ovat nähtävissä ja liikuteltavissa. Käyttäjä tarvitsee osoitinlaitteen, jolla hän voi kontrolloida osoitinta käyttöliittymässä. Yleisin osoitinlaite työpöytä tietokoneissa on erillinen hiiri. Tämän tyyppinen osoitinlaitteen käyttö ei kuitenkaan ole kovinkaan ergonomista, sillä käyttäjän käsi joutuu olemaan huonossa asennossa pitkiäkin aikoja. Lisäksi erillinen hiiri olisi erittäin epäkäytännöllinen esimerkiksi kännykän tai tabletin osoitinlaitteeksi. Näissä päätelaitteissa onkin useimmiten kosketusnäyttö, joka toimii joko käyttäjän sormen kosketuksella, tai erillisellä Stylus-kynällä. (Jaakkola 2011)

Vaikka hiiri on pitänyt pintansa osoitinlaitteiden kärjessä jo yli 50 vuotta, ja sen ominaisuuksia parannellaan jatkuvasti, on todennäköistä, että tekniikan koko ajan kehittyessä ottaa joku innovatiivisemmista ratkaisusta pian hiiren valta-aseman. Hiiren dominointi osoitinlaitteena on myös johtunut osittain siitä, että käyttöliittymät ovat suosineet sitä. Nyt kuitenkin esimerkiksi Windows 8:n tultua markkinoille, kosketusnäyttö myös pöytäkoneissa on varsin huomioon otettava vaihtoehto. Osoitinlaitteiden kehityksessä on peliala suunnannäyttäjänä. Markkinoilta löytyy jo muun muassa pään- ja silmien liikkeen tunnistava ohjainlaite, ”datahanska”, joka tunnistaa käyttäjän eleet, sekä ratti autosimulaattoreihin. Vaikka puheentunnistukseen tarkoitettuja laitteita ja ohjelmia on tullut jo jonkin verran markkinoille, on näppäimistö kuitenkin tekstin tuottamisen välineenä vielä ylitse muiden. Myös monissa kännyköissä on vielä pelkästään näppäimistö syöttölaitteena.



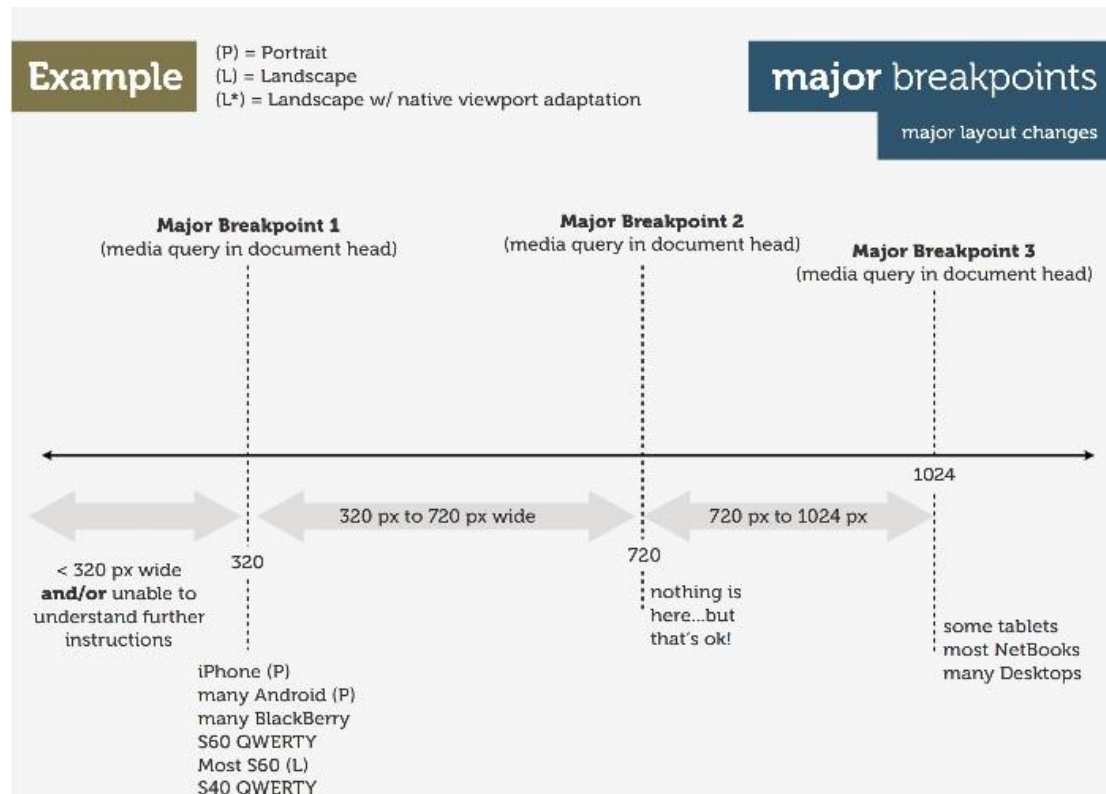
**KUVA 2. Joustava mininäppäimistö (Wireless flexible keyboard 2012)**

### 3 RESPONSIIVINEN VERKKOSUUNNITTELU

Responsiivisen suunnittelun keskeisimpiä termejä on "joustavuus". Sivun elementit suunnitellaan niin, että ne mukautuu erikokoisille päätelaitteille, sillä emme voi mitenkään olla varma minkälaisella alustalla julkaisuamme tarkastellaan. Emme myöskään voi suunnitella jokaiselle alustalle omaa versiota, sillä erilaisia resoluutioita käytäviä laitteita on yksinkertaisesti liian monta (jokaiselle sovellukselle tulisi olla myös oma ylläpito, mikä tarkoittaisi massiivisia kustannuksia). Joissain tapauksissa on järkevää tehdä isoille resoluutioille, sekä pienille resoluutioille (kännykät) omat sovellukset, mutta yleisesti ottaen kehitystyön tulisi alkaa menemään enemmän responsiivisen suunnittelun suuntaan. (Macrotte 2010, 4-7)

Responsiivinen suunnittelu ei tarvitse mitään uutta ohjelmointikieltä tai uusia välineitä, vaan se keskittyy nimenomaan käyttämään olemassa olevia välineitä, kyse on vain paremmasta suunnittelusta ja ohjelmoijan halusta tehdä laitteesta/näytöstä riippumattonta sisältöä. Tällaisen ajattelun suurimpia etuja on se, että sivustoista tulee paremmin aikaa kestäviä. Nykyisin enemmän käytettävässä tyylissä sivut suunnitellaan tietylle resoluutiolle (esim 420px) ja heti kun tämä ei enää olekaan standardiresoluutio esimerkiksi kännyköissä, joudumme suunnittelemaan sivumme uudestaan. Parhaan tuloksen responsiivisille sivuille saa, kun ne tekee alusta asti *fluid grid* pohjalle ja ottaa muutenkin responsiivisen näkökulman heti suunnitteluvaiheessa mukaan. Myös valmiiden sivustojen, joita ei ole suunniteltu responsiiviseksi, responsiivisuutta voidaan parantaa jälkikäteen, mutta se on huomattavasti työläämpi prosessi. Verkkosovellukselle luodaan tiettyjä *breakpointeja*, eli käytännössä resoluutioehtoja, joiden täytyessä sovelluksen ulkoasu mukautuu kullekin resoluutiolle (tai käytettävän laitteen näytön leveyden mukaan) sopivaksi. Näistä breakpointeista pitää tehdä arvio suunnitteluvaiheessa, ja niihin vaikuttavat minkälaisilla laitteilla käyttäjät oletettavasti tulevat vierailemaan sivuilla. Useimmissa tapauksissa riittää, kun katsoo esimerkiksi myydyimpien kämmenlaitteiden tilastoja ja selvittää niiden resoluutiot. Usein näille breakpointeille kannattaa tehdä omat tyylitiedostot, jotta esimerkiksi mobiilikäyttäjät, joilla saattaa olla hitaampi verkkoyhteys tai tiedonsiirtomaksut, eivät joudu lataamaan turhaa sisältöä, jota he eivät kuitenkaan näe. Näihin tyylitiedostoihin voidaan esimerkiksi määritellä omat (pienemmät) kuvatiedostot, jotta mobiilikäyttäjät eivät joudu

turhaan lataamaan suuria kuvatiedostoja, jotka sitten kutistetaan esimerkiksi kännykän näytölle sopiviksi. Nämä breakpointit eivät kuitenkaan pelkästään riitä, sillä ne mukautuvat vain näytön tai selaimen leveyden mukaan, eivätkä ota huomioon päätelaitteen muita ominaisuuksia. Tästä syystä responsiivinen suunnittelu usein tarvitsee parikseen selaimen ominaisuuksien tunnistuksen ja näihin ominaisuuksiin sopeutumisen (Rieger 2011). Kuvassa 3 nähdään kolme hyvin yleisesti käytettyä breakpointia.



**KUVA 3. Esimerkkejä mahdollisesti tärkeistä breakpointeista (Rieger 2011)**

### 3.1 One web ja mobile first

Internetin ajatus oli alun perin olla avoin kaikille ja mahdollisimman monenlaisille sivustoille. Jeremy Keithin lanseeraama konsepti "one web" haluaa tukea tätä ajatusta. Nykyään tehdään erilaisia natiiveja, sovelluksia tietynlaisille alustoille/päätelaitteille (niinkuin iOS), joilla halutaan ottaa tietyn alustan täysi kapasiteetti käyttöön. Kuitenkin "one web" konseptin luojan Jeremy Keithin mielestä iOS:n paras sovellus on selain! Tätä avoimuutta tukeakseen sivustojen suunnittelijoiden on alettava ajatella responsiivisesti. Kysymys kuuluukin, miksi haluaisit lukita sisältösi yhden tietyn tyyppi-

sen laitteen/näyttöpäätteen käytettäväksi, jos voit tehdä sovelluksestasi responsiivisen. Useimmat vastaväitteet responsiivista- ja *one web*-ajattelua vastaan koskevat tiedon määrää, "*sivusto on liian laaja kännyköille*", tai "*siinä on liikaa tietoa nopeasti etsittäväksi*", tämä on totta, ja tästä syystä mukaan on otetta myös *mobile first* -ajattelumalli. Mobile first - ajattelumallissa lähdetään ajatuksesta, jos tietoa ei tarvitse näyttää mobiilipäätteellä, ei kyseessä oleva tieto ole myöskään tärkeää työpöydän selaimella näytettäväksi. Tämä ajatusmalli tuottaa siis yksinkertaisempia sivustoja, joissa oleellinen tieto on helposti löydettävissä. Mobile first - ja oneweb -ajattelumallit ovat totta kai syntyneet tarpeeseen, sillä älypuhelimien ja tablettien-, sekä niillä suoritettavan data-liikenteen määrä on huimassa kasvussa.

Tim Berners-Lee kiteyttää *One web:n* tärkeyden hienosti Scientific Americanin artikkelissaan *Long Live Web: A Call for Continued Open Standards and Neutrality*, "*Webin tulisi olla käytettävissä ilman rajoituksia. Sen täytyisi toimia kaikenlaisen tiedon alustana, oli se sitten dokumentti tai nippelitieto – sekä oli tiedonlaatu sitten mikä hyvänsä – simppeleistä tweetistä tieteellisiin papereihin. Sen pitäisi myös olla saatavissa kaikenlaisilta laitteilta, joilla pääsee internetiin, staattisilta, mobiileilta,, pieniltä ja suurilta näytöiltä jne* ” (Berners-lee 2010).

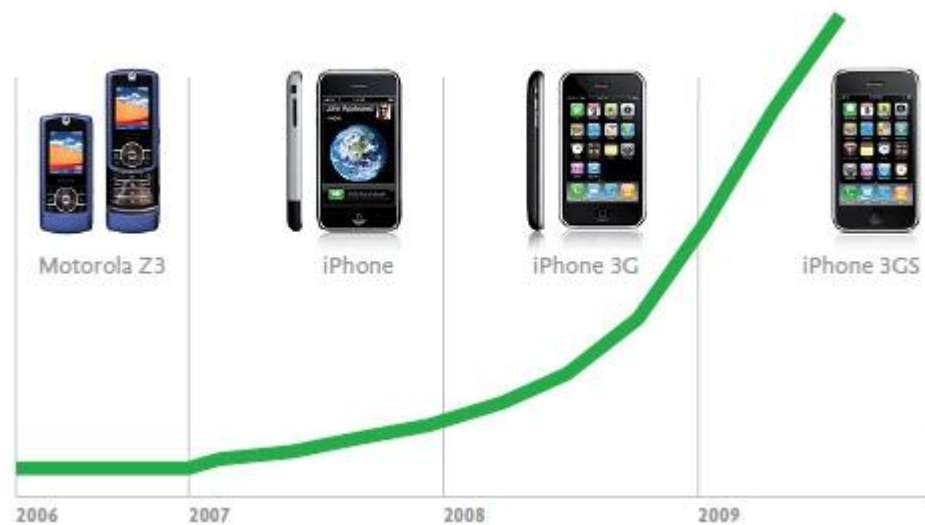
Lähes tulkoot kaikki tunnustavat *one webin* ideaalitalanteeksi, vastaväitteitä tulee vain siitä, että onko se mahdollista tai kuinka siihen suuntaan päästään. Tähän asti tulleista ratkaisusta parhaaksi on osoittautunut nimenomaan responsiivinen verkkosuunnittelu yhdistettynä mobile first -ajatteluun (Keith 2010).

### **3.1.1 Mobiililaitteiden verkon mahdollisuudet**

Kuvasta 4 voidaan todeta kaksi asiaa, mobiililaitteiden dataliikenteen hurjan nousun, sekä Applen iPhoneen vaikutuksen siihen. On erittäin todennäköistä, että dataliikenteen määrä tulee myös lähivuosina kasvamaan, jo nyt noin miljardi ihmistä käyttää mobiiliverkkoa aktiivisesti. Mobiiliverkossa tutustutaan tuotteisiin ja palveluihin, tehdään ostoksia ja verkostoidutaan. Tämä asetelma luo yrityksille aivan uudenlaisia mahdollisuuksia, mutta myös riskejä. Mobiiliverkon käytöllä voidaan nopeuttaa työn tekoa, ja parantaa tuottavuutta yrityksessä. Lisäksi tiettyjen mobiilipäätelaitteiden hin-

nat, kuten vanhempien älypuhelimien, ovat paljon alhaisempia kuin perinteisen työpöytäkoneen, joten voidaan tavoittaa potentiaalisia asiakkaita, joilla ei välttämättä aiemmin ollut varaa toimia verkossa. Mobiiliverkon riskit liittyvät yleensä tietoturvaan, ja tarkemmin ottaen langattomiin lähiverkkoihin. Yritys ei välttämättä voi valvoa mitä ohjelmia työntekijät älypuhelimiinsa lataavat, tai mitä verkkoa he käyttävät vapaa-ajallaan, ja että onko nämä turvallisia. Lisäksi tietojen varmuuskopiointi on haastavampaa mobiililaitteilla verrattuna perinteisiin pöytäkoneisiin.

(Wroblewski 2011, 6-14)



**KUVA 4. Dataliikenteen kasvua Amerikassa, iPhonen tultua markkinoille (Wroblewski 2011, 11)**

### 3.1.2 Natiivit sovellukset vs. mobiiliverkko sovellukset

Natiiveilla sovelluksilla tarkoitetaan sovelluksia, jotka ovat suoraan asennettuna päätelaitteelle. Natiivisovellus on suunniteltu varta vasten tietylle päätelaitteelle ja käyttäjät lataavat niitä yleensä laitteen omasta verkkokaupasta, kuten Appstore tai Google play. Latauksen ja asennuksen jälkeen natiivia sovellusta voidaan käyttää myös offline-tilassa. Mobiiliverkkosovellukset puolestaan ovat mobiililin päätelaitteen selaimella tutkittavia sovelluksia, joita ei tarvitse erikseen ladata ja asentaa. (Six Revisions 2012)

Jos verkkopalvelu suunnitellaan alusta alkaen mobile first ja responsiivisen suunnittelun malleja noudattaen, ei erillistä natiivisovellusta usein tarvita. Uusimmat selaimet mobiililaitteissa tukevat muun muassa paikannustunnistusta ja laitteen suuntaustietoja. Toisaalta joitain ominaisuuksia, kuten videokameraa ja magnetometriä, mobiiliselain ei yksinkertaisesti pysty tarjoamaan, jos palvelusi pystyy hyödyntämään näitä, kannattaa valita natiivisovellus. (Wroblewski 2011, 34-35)

Mobiiliverkkosovelluksen, joka on suunniteltu responsiivisesti, puolesta puhuu myös sen hyöty-kustannus suhde useimmissa tapauksissa. Responsiivinen verkkopalvelu tarvitsee vain yhden ylläpidon, ja uusia mahdollisesti kalliita päivityksiä ei tarvita, kun markkinoille tulee uusi mobiililaitte. (Scaglione 2012)

### 3.2 CSS3

Vuonna 1994 verkkojulkaisut alkoivat yleistyä: ongelmana kuitenkin oli, ettei julkaisijoilla ollut mitään menetelmiä tehdä julkaisulleen tyyliä dokumentteja. Tyylien erottaminen sisällöstä ei ollut uusi idea, sillä se oli ollut HTML:n tähtäimenä alun perinkin. CSS (Cascading Style Sheets) tarjosi ratkaisun tähän ongelmaan. Tarjolla oli myös muitakin ratkaisuja, mutta CSS:n pyrkiminen ottamaan huomioon julkaisijan, julkaisun lukijan, sekä selaimen ja päätelaitteen ominaisuudet, nostivat sen parhaaksi vaihtoehdoksi.

Rakenteen ja tyylien erottaminen toisistaan mahdollistaa erilaisten layouttien luomisen eri päätelaitteille sisällön pysyessä kaikille samana. Yleisimmin sitä käytetään (x)HTML:n kanssa, mutta ne eivät ole toisistaan mitenkään riippuvaisia. Vuonna 1996 W3C julkaisi ensimmäisen spesifikaationsa CSS:stä. Samana vuonna julkaistiin Internet Explorer 3.0 selain, jolla ei kuitenkaan ollut kunnollista tukea CSS:lle. CSS:n tuki selaimilla onkin ollut vaihtelevaa sen alkuajoista lähtien, vaikka esimerkiksi Netscape 4 pyrki tarjoamaan tuen CSS:lle, ei selaimen kehittäjillä ollut tarpeeksi aikaa testata kaikkia ominaisuuksia, ja tuki jäikin vajaaksi ja bugeja ilmeni paljon. Bugisuus oli riesana myös Internet Explorerin uudemmissa yrityksissä tukea CSS:sää. (BlizzAdmin 2008)

Vuonna 2003 julkaistiin safariselain ja vuonna 2004 Mozilla Firefox, jotka molemmat olivat rakennettu tarkemmin noudattamaan standardeja. Näissä selaimissa oli laaja

CSS tuki, ja bugeja ilmeni huomattavasti vähemmän. Internet Explorer on kuitenkin edelleen maailman suosituin selain (netmarketshare), ja myös sen vanhempia versioita käytetään edelleen. Tästä syystä verkkosuunnittelijoiden on hyvä käyttää erilaisia *CSS-filttereitä* ja ”*workarundeja*”, joilla pyritään saavuttamaan verkkosivujen yhtenäinen ilme selaimesta riippumatta. (The CSS saga 2012)

CSS:n uusin versio CSS3 on rakennettu CSS 2.1:n päälle erillisistä moduuleista. Tällainen moduulipohjainen rakenne auttaa nopeuttamaan W3C spesifikaatioiden ja suositusten laatimista ja hyväksymistä, sillä ne voidaan tehdä moduulikohtaisesti. Lisäksi selaimet voivat tukea esimerkiksi tiettyjä moduuleita, eikä tarvitse tukea koko CSS3 pakettia (<http://www.css3.info/modules/> sivusto pitää kirjaa moduuleista, jotka esiintyvät W3C spesifikaatiossa).

Aikaisemmin mainitun selaintuen puutteen vuoksi, esimerkiksi CSS3:n tarjoamaa kolmea layout moduulia ei kannata toistaiseksi käyttää. Muutenkin CSS3:n käyttämistä tällaisiin verkkosivujen toiminnan kannalta kriittisiin toimintoihin kannattaa toistaiseksi välttää. Toisaalta taas, visuaalista ilmettä ja käyttäjäkokemusta rikastavia ominaisuuksia voidaan jo tarjota sillä selaimet yksinkertaisesti ohittavat nämä CSS3 määrittymiset, mikäli niitä ei tueta. Tällaisia CSS3:n tarjoamia uusia ominaisuuksia ovat muun muassa *border-radius*, *text-shadow*, *box-shadow*, *multiple background images*, *opacity* ja *RGBA*. Näistä ominaisuuksista esimerkiksi *Border-radius* pyöristää elementin kulmia tietyn säteen verran, ja se merkattaisiin CSS tiedostoon seuraavasti:

```
.foo {  
-webkit-border-radius: 10px;  
-moz-border-radius: 10px;  
border-radius: 10px;  
}
```

Kuten merkinnästä huomataan, jotkin selaimet tarvitsevat etuliitteen, jotta ominaisuus toimii oikein. Ominaisuus merkataan myös lopuksi ilman etuliitettä, koska kaikki selaimet eivät sitä tarvitse. Näin merkattuna se on myös paremmin suojattu tulevaisuuden varalle, sillä selaimet käyttävät aina viimeisenä merkattua ehtoa, mikäli sitä on tuettu. (Zeldman 2010, 4-12)

Yksi etenkin responsiivisen suunnittelun kannalta tärkeä uusi CSS3 ominaisuus on mediakyselyt. W3C on jo pitkään miettinyt ratkaisua miten verkkosivujen tulisi muokautua erikokoisille ja muotoisille päätelaitteille. CSS2:ssa ratkaisuksi tarjottiin *media*luokkia, joissa määriteltiin minkälaiselle päätelaitteelle tyyli ja layout on tarkoitettu. Tuettuja media-*luokkia* oli muun muassa *all*, *handheld*, *print*, *screen* ja *tv*. Ongelmana kuitenkin oli, että esimerkiksi *handheld* - media-*luokka* oli liian laaja käsite kattamaan kaikki kännykät, niiden näytön kokoerojen johdosta. Lisäksi kännykän selaimissa ei pitkään aikaan ollut kunnon tukea media-*luokille*. CSS3:ssa julkaistu mediakyselyt yrittää tarjota ratkaisun tähän ongelmaan. Se on eräänlainen testi päätelaitteelle tai selaimelle, jolla tutkitaan sen fyysisiä ominaisuuksia. Jos haluamme esimerkiksi testata päätelaitteen leveyttä, merkataan tyyliin media-*luokan* nimi ja sen jälkeen annetaan joko maksimi - tai minimiarvo, eli minkä leveyisellä laitteella tätä tyyli-tiedostoa halutaan käyttää, esimerkiksi:

```
@media screen and (min-width: 1024px) {  
  body {  
    font-size: 100%  
  }  
}
```

Nämä *min-width* tai *max-width* arvot toimivat myös *responsiivinen verkkosuunnittelu* kappaleessa mainittuina *breakpointeina*. (Macrotte 2010, 72-75)

Kuvassa 5 nähdään, minkälaisia testejä voimme asettaa mediakyselyillä päätelaitteelle leveyden ja korkeuden lisäksi.



FEATURE NAME	DEFINITION	HAS min- AND max- PREFIXES
orientation	Accepts portrait or landscape values.	✗
aspect-ratio	Ratio of the display area's width over its height. For example: on a desktop, you'd be able to query if the browser window is at a 16:9 aspect ratio.	✓
device-aspect-ratio	Ratio of the device's rendering surface width over its height. For example: on a desktop, you'd be able to query if the screen is at a 16:9 aspect ratio.	✓
color	The number of bits per color component of the device. For example, an 8-bit color device would successfully pass a query of (color: 8). Non-color devices should return a value of 0.	✓
color-index	The number of entries in the color lookup table of the output device. For example, @media screen and (min-color-index: 256).	✓

**KUVA 5. Mediakyselyn ominaisuudet (Macrotte 2010, 77)**

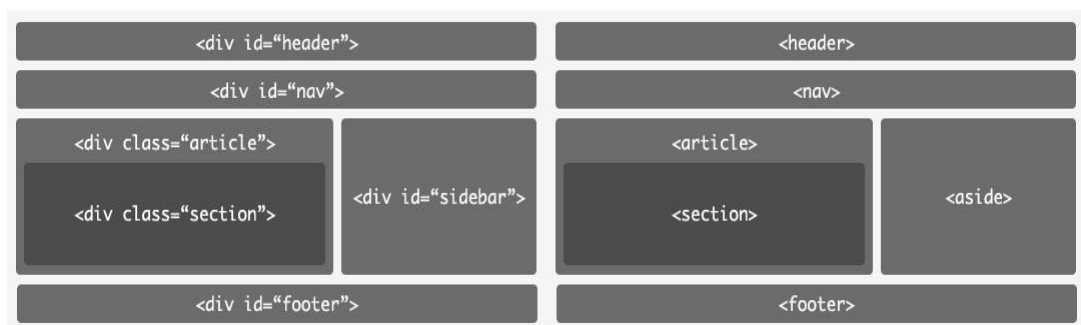
### 3.3 HTML5

HTML eli *hypertext markup language* on word wide webin yhdistävä, W3C standardoitu kieli. Sitä on kehitetty jatkuvasti siitä lähtien, kun Sir Tim Berners-Lee esitteli ideansa vuonna 1991 tekstissään ”HTML Tags”. Kieltä kehitettiin W3C:n valvonnassa suuremmista ongelmista HTML 4.01:n julkaisuun, eli vuoteen 1999 asti. Tällöin julkaistiin päivitetty versio nimeltä XHTML, jossa kielen syntaksia oli muutettu. Standardeja tiukennettiin huomattavasti, ja suunnittelijoiden tuli noudattaa XML:n tiukempia merkintäsääntöjä. Ongelmat alkoivat kun XHTML 1.1 julkaistiin, sillä sen hetkisistä selaimista ylivoimaisesti suosituin, eli Internet Explorer ei tukenut sitä.

W3C ei silti ollut valmis luovuttamaan, vaan jatkoi kehitystyötä webin XML-pohjaista tulevaisuutta kohti ja aloittivat rakentamaan XHTML 2:sta. Prosessi oli fiasko, sillä XHTML2:lla ei ollut yhteensopivuutta aiemmin julkaistun verkkosisällön tai edes HTML:n kanssa. Vaihtoehtoja XHTML2:lle kaivattiin nopeasti, ja Ian Hickson kollegoineen perustikin WHATWG (the Web Hypertext Application Technology Working Group) ryhmän, joka alkoi kehittää Web Forms 2.0 ja Web Apps 1.0:aa HTML:n laajennukseksi, jotka myöhemmin yhdistyi HTML5 nimen alle. Myös W3C hyväksyi projektin sen jälkeen, kun Sir Tim Berners-Lee kirjoitti blogissaan vuonna 2006, ettei webiä kannattaisi ajaa HTML:stä XML-pohjaiseksi. W3C jatkoi silti myös XHTML2:n kehitystyötä aina vuoteen 2009 asti.

Ian Hickson kertoi haastattelussa, että HTML5 olisi valmis vuoteen 2022 mennessä, tämä ei kuitenkaan tarkoita, ettei sitä voisi jo käyttää tänään. Paljon uusia ominaisuuksia on tarjolla, ja tärkeämpää kuin HTML5:n ”valmistumisvuosi” onkin selainten tuki näille ominaisuuksille. Täydellistä selaintukea ei vielä ole, mutta kun sen ottaa verkkosuunnittelussa huomioon ja käyttää HTML5 elementtejä sillä tavalla, että vaikka selain ei niitä tukisi, on sivusto silti käyttäjäystävällinen ja uusia elementtejä turvallis- ta käyttää.

HTML5 tarjoaa reilusti uusia rakennetta selkeyttäviä semanttisia elementtejä, sekä myös sisältöä tarjoavia elementtejä. Näitä uusia semanttisia elementtejä ovat muun muassa *section*, *nav*, *article*, *header*, *footer* ja *aside*. Hakukoneet ja selaimet saavat enemmän tietoa näistä uusista elementeistä, ja niiden käyttö voi muun muassa parantaa verkkosivun hakukone sijoittumista huomattavasti. Kuvassa 6 nähdään vanha ja uusi merkintätapa HTML rakenteesta.



**KUVA 6. Vanha ja uusi tapa rakentaa HTML-dokumentti (Jaakkola 2011)**

Sisältöä tarjoavia uusia elementtejä ovat muun muassa *video*, *audio* ja *canvas*. Video- ja audio elementeillä voidaan toistaa kyseisiä medioita natiivisti selaimessa, ja canvas tarjoaa piirtoalustan, jonka avulla voidaan esimerkiksi piirtää grafiikkaa reaaliaikaisesti selaimella. Kaikilla näillä kolmella elementillä pyritään pienentämään Flash-tekniikan ja vastaavien selaimen lisäosien tarvetta. (Keith 2010)

HTML5:een on integroitu alun perin W3C:n erillisenä spesifikaationa ollut *Web forms 2.0* ja HTML5:n lomakkeet sisältävätkin useita tämän spesifikaation ominaisuuksia. Kuten aikaisemmassa luvussa mainituista sisältöä tarjoavissa elementeissä, myös HTML5 lomakkeissa on pyritty vähentämään erillisten kirjastojen ja selaimen lisäosien tarvetta. Uusia elementtejä ovat muun muassa *calendar*, *search*, *time*, *email*, *date*, ja *url*. Nimenomaan kalenteri onkin ollut yksi JavaScript kirjastojen suosituimmista ominaisuuksista, ja nyt HTML5:n tarjotessa sen natiivina vaihtoehtona, sen lisäksi, että erillistä lisäosaa ei tarvita, myös käyttäjien käyttökokemuksista tulee yhtenäinen. Nämä syöttökentät käyttäytyvät normaalin tekstikentän tapaan, mikäli selaimelta ei löydy HTML5 tukea. Uusista attribuuteista voidaan mainita ainakin *placeholder*, *autofocus*, *required*, sekä *autocomplete*. Placeholder on lomakkeelle valmiiksi syötettyä tekstiä, joka antaa käyttäjälle vihjettä siitä minkälaista - ja missä muodossa tietoa tulisi kenttään syöttää. Autofocus kohdistaa kursorin tiettyyn lomakkeen kenttään. Required attribuutilla voidaan määritellä pakolliset kentät, joka estää lomakkeiden lähetyksen, mikäli kyseiset kentät ovat jätetty tyhjiksi. Autocomplete muistaa käyttäjän aikaisemmin lomakkeelle täyttämiä tietoja ja täydentää lomaketta automaattisesti näiden tietojen perusteella.

Selaimien tuen puutteen vuoksi näiden attribuuttien käytön kanssa kannattaa olla vielä todella varovainen. Miellyttävän käyttäjäkokemuksen ja tietoturvan saavuttamiseksi, kannattaakin näiden käyttöä harkita tarkkaan, ja käytettäessä laittaa myös toiminto JavaScriptinä varmistukseksi. (Jaakkola 2011)

### 3.4 Joustava sommittelu

Kuten *responsiivinen verkkosuunnittelu* kappaleessa todettiin, on parhaan responsiivisuuden saavuttamiseksi tärkeää, että verkkosivut toteutetaan alusta alkaen joustavalle pohjalle. Joustavan sommittelun saavuttamiseksi, Ethan Macrotte ,2010, esitteli kirjas-

sa responsive web design Target/Context=result kaavan, jolla muun muassa fontit, palstat, marginaalit, rajaukset ja paddingit voidaan esittää paremmin päätelaitteelle mukautuvassa muodossa käyttäen pikseleiden sijaan prosentuaalisia ja suhteellisia arvoja. Jos esimerkiksi olisimme suunnitelleet layoutin 960 pikselin levyiseksi, joka on standardi tällä hetkellä, ja haluaisimme siihen kaksi palstaa leveydeltään 566px ja 331 px (tässä esimerkissä ylimääräinen osa jätetään tyhjää tilaa sivun reunoille), voisimme käyttää target/context = result kaavaa ja merkatakin palstojen leveyden prosentteina.

Eli laskukaava näyttäisi tältä:

$$566/900 = 0.628888889 = 62.8888889\%$$

$$331/900=0.367777778 = 36.7777778\%$$

ja css merkintä tältä:

.ensimmäinen palsta

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

.toinen palsta

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

Samaa laskentakaavaa voidaan siis käyttää myös esimerkiksi paddingeihin ja marginaaleihin, mutta suhteellisia arvoja laskettaessa, tulee huomioida että paddingeilla konteksti on elementti itsessään, ja marginaaleilla konteksti on elementin säiliö (Div). Fontit merkataan joustavassa sommittelussa suhteellisilla arvoilla em, jossa 1em vastaa selaimen oletus fonttikokoa (yleensä työpöytäkoneiden selaimella 16px), jos siis haluaisimme luoda esimerkiksi 24pikselin kokoisen otsikon, se merkattaisiin joustavammalla merkinnällä  $24/16 = 1.5$  eli: `h1 { font-size: 1.5em;}`. (Macrotte 2010, 19-35)

Joustava sommittelu on hyvä perusta responsiivisille verkkosivuille, mutta se ei kuitenkaan yksin riitä responsiivisuuden saavuttamiseksi, sillä se ei taivu näyttölaitteiden kokoeron kasvaessa hyvin paljon. Tällöin tarvitaan aikaisemmassa kappaleessa mainittu *breakpointti*, jossa layout muuttuu paremmin kyseessä olevalle kokoluokalle sopivaksi. Kun kaikki nämä layoutit ovat joustavasti sommiteltu ja breakpointeja määriteltä kriittisiin kohtiin, ollaan jo hyvin lähellä responsiivisten verkkosivujen saavuttamista. (CSS Units 2012)

### 3.5 Joustavat kuvat

Yleisesti käytössä olevalla tyylillä pixleinä ilmoitettu kuvan korkeus ja leveys on täysin joustamaton, eikä sovi responsiivisen verkkosuunnittelun malliin. Käyttämällä CSS3:n max-width ominaisuutta, kuva skaalautuu aina sille tarkoitetun säiliön kokoiseksi, paitsi silloin jos kuva on alun perin pienempi kuin säiliö, silloin kuva pysyy oletuskoossaan. Max-width ominaisuus toimii myös muille mediaelementeille, kuten *video*, *object* ja *embed*. Max-width ominaisuudella on varsin kattava selaintuki, mutta IE6 ja vanhemmat IE:n versiot eivät sitä tunnista. Näiden vanhempien selainten varalle voidaan asettaa width 100 % arvo, joka venyttää kuvan *aina* säiliönsä kokoiseksi. (Bradley 2011)

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

ja IE:tä varten

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

Taustakuville CSS3 tarjoaa background-size ominaisuuden, joka sopisi myös responsiivisille sivuille mainiosti, mutta selaimet eivät valitettavasti vielä tue tätä ominaisuutta. Jotta saisimme taustakuvista joustavampia, voimme käyttää *joustavat sommitelut* -kappaleessa esiteltyä target/context=result kaavaa.

Lasketaan alkuperäisestä sommittelusta saatujen tietojen mukaan prosentuaalinen arvo sille missä kohdassa väri vaihtuu tummemmaksi:

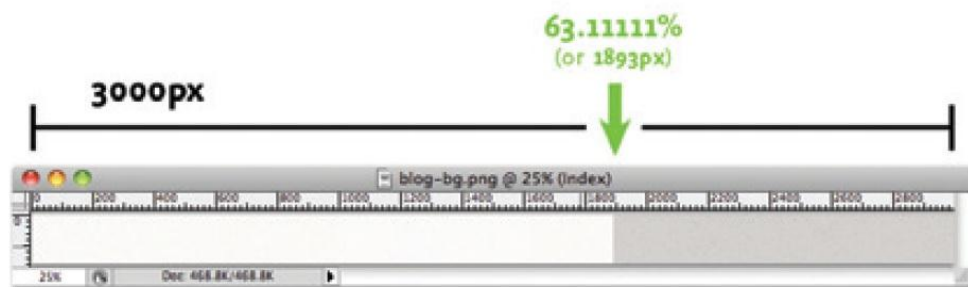
$$568/900=0.631111111 = 63.1111111\%$$

(Macrotte 2010, 42-50)



**KUVA 7. Esimerkin alkuperäinen taustakuva (Macrotte 2010, 56)**

Tehdään esimerkiksi Photoshopilla yllä olevä palkki, tässä tapauksessa 3000 pikseliä, ja asetetaan värin vaihto aikaisemman saadun prosentuaalisen arvon perusteella.



**KUVA 8. Leveä palkki responsiivisempaa taustaa varten (Macrotte 2010, 57)**

Asetetaan tämä kuva taustakuvaksi ja toistetaan sitä pystysuunnassa. Nyt taustamme skaalautuu pienistä kännykän näytöistä aina 3000 pikseliin asti.

```
.palsta { background: #F8F5F2 url("taustakuva.png") repeat-y » 63.11111111111111%
0; }
```

(Macrotte, 2010, 54-56)

#### 4 CASE: PORTFOLIOT

Työnäytteitä sisältävä portfolio, on nykyään työmarkkinoilla usein vaadittava dokumentti perinteisen CV:n lisäksi. Etenkin digitaalisilla aloilla, *Eportfolio* eli eräänlainen sähköinen ansioluettelo alkaa jo löytyä yhä useammalta työnhakijalta. Eportfolion luomiseen ja ylläpitämiseen on tarjolla erilaisia palveluja, kuten LinkedIn ja VisualCV, näissä palveluissa käyttäjä voi myös hoitaa verkostoitumista oman alansa ihmisten kanssa. Tässä opinnäytetyössä haluttiin saada mukaan responsiivisen suunnit-

telun, sekä mobile first ajattelumallin hyödyt, joka ei ole mahdollista edellä mainituissa palveluissa, joten Eporfoliot toteutettiin omille verkkosivuilleen.

#### **4.1 Portfolioiden vaatimusmäärittely ja suunnittelu**

Vaatimuksena oli toteuttaa kaksi responsiiviseksi suunniteltua portfolioa, toinen teolliselle muotoilijalle ja toinen minulle itselleni. Aloitin suunnittelemalla portfolioita mielessäni, että niiden tulisi olla selkeät, sekä käyttäjäystävälliset. Niiden tulisi myös näyttää hyvältä, sekä säilyttää samanlainen tunnistettava ilme, sekä työpöytäkoneissa, kännyköissä, kuin tableteillakin. Työssä tulisi näkyä myös mobile first - ajattelumalli, jossa verkkosivuille ei ahdeta liikaa sisältöä, eli kaikki ylimääräinen karsitaan pois. Toisessa portfolioissa tulisi myös ottaa huomioon asiakkaan toiveet ulkoasun suhteen. Toteutuksen vaatimuksena oli, että työssä käytetään HTML5:sta ja CSS3:sta, Javascriptiä ja sen tapaisia koodeja käytetään mahdollisimman vähän, sekä erinäisiä julkaisujärjestelmiä vältetään kokonaan, eli työt toteutetaan niin sanotusti puhtaalta pöydältä.

Aloitin suunnittelutyön keskustelemalla teollisen muotoilijan Minna Ukkosen kanssa, hänen toiveistaan portfolion suhteen. Esille tulivat ainakin asiat, että portfolion tulisi sisältää paljon kuvia, jotka toimivat työnäytteinä, hieman tietoa Minnasta itsestään, sekä tyylin tulisi olla selkeä ja hieman retrohenkinen.

Luonnostelin ideoitani portfolion ulkoasusta Photoshopilla esiteltäväksi Minnalle, prosessissa käytettiin ketteriä menetelmiä, sillä minulla oli mahdollisuus jakaa mielenpitoita Minnan kanssa päivittäin ja käydä ulkoasua (ja myöhemmin toiminnallisuutta) läpi kohta kohdalta. Kun työpöytäkoneen ulkoasusta päästiin yhteisymmärrykseen, laadin myös ehdotukseni kännykkä ja tabletti näkymiksi, jotka myös hyväksyttiin pienen hienosäädön jälkeen. Kuvassa 9 nähdään alkuperäiset layout suunnitelmat.



**KUVA 9. Alkuperäiset layout suunnitelmat**

Omassa portfolioissani jätin tämän työvaiheen välistä, enkä laatinut lainkaan Photoshop luonnoksia, vaan sain toteuttaa oman tyylliset sivut vapaalla kädellä. Otin mukaan myös ideoita joistakin valmiista responsiivisiksi verkkosivuista.

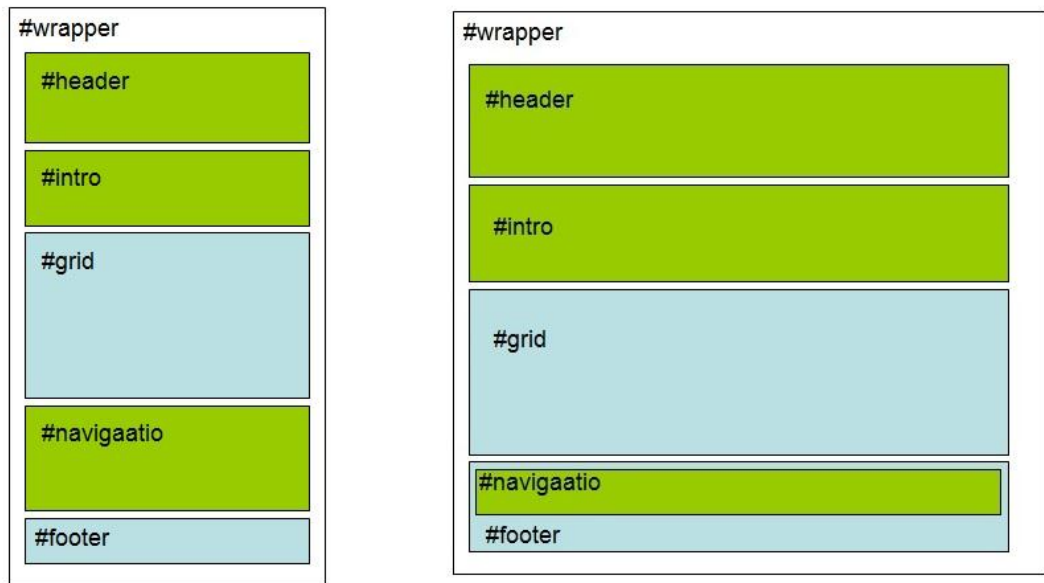
Mobile first ajattelumalli tuli esille molemmissa töissä niin, että sivuille ei ahdettu ylimääräistä tietoa, eli kaikki tieto oli saatavissa nopeasti ja helposti myös pienemmillä näyttöpäätteillä. Koska työssä haluttiin esitellä myös CSS3:n ja HTML5:n uudempiä ominaisuuksia, joita esimerkiksi kaikkien mobiililaitteiden selaimet eivät tue, oli vaatimuksena suunnitella esimerkiksi yhteydenottolomakkeet niin, että uusia ominaisuuksia saatiin esiteltyä, mutta että ne olisivat myös käyttäjäystävälliset selaimilla, jotka eivät vielä tue ko. ominaisuuksia.

#### **4.2 Käytännön toteutus (portfolio1)**

En ollut aikaisemmin suunnitellut tai toteuttanut responsiivisia verkkosivuja, mutta tämän opinnäytetyön teoriaosuuden kirjoitettuani ja aiheesta paljon luettuani tiesin, että tärkeimmät tekijät onnistuneen toteutuksen kannalta olisivat joustava sommittelu, joustavat kuvat ja mediakyselyt. Laadin sivujen pohjaksi HTML5 dokumentin (merkaataan <!DOCTYPE html>), joka sisälsi jo yleiseksi standardiksi muodostuneet tagit,



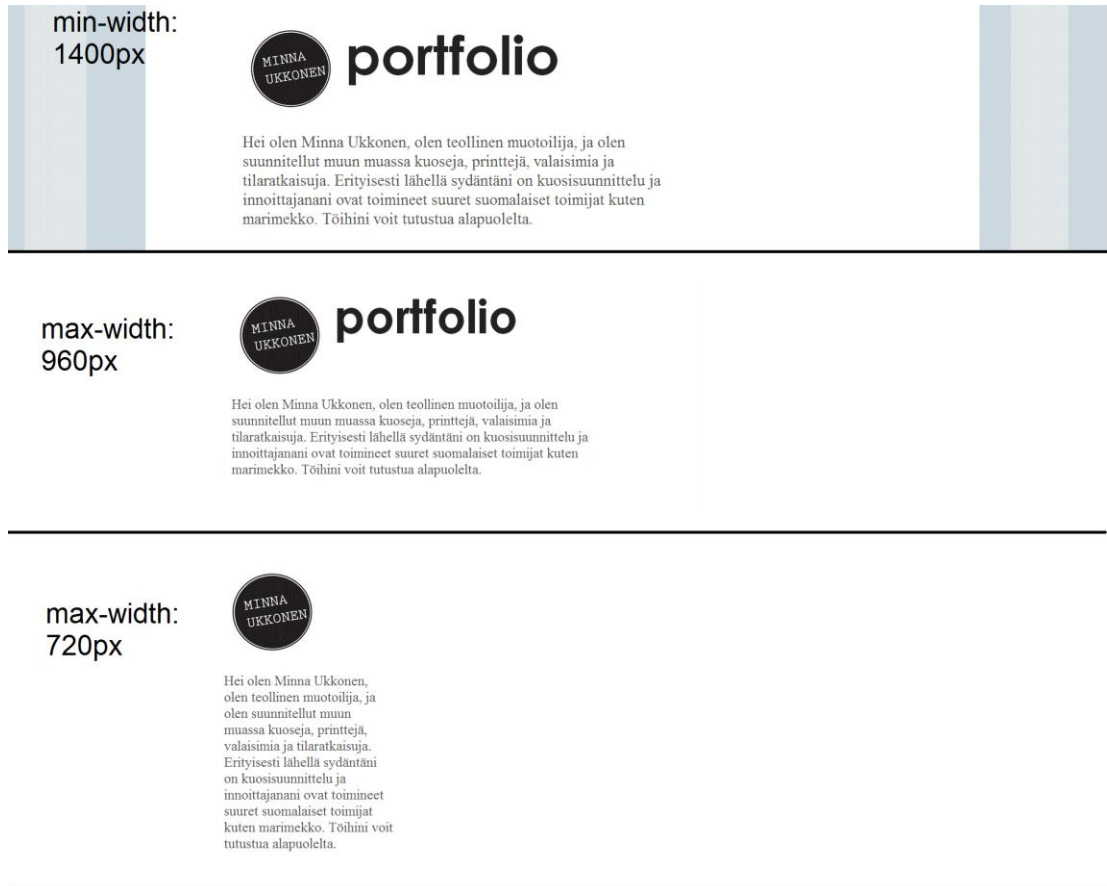
kuten <html>, <head>, ja <body>. Body tagin sisään tein div elementtejä, joiden ominaisuuksia myöhemmin tyylitiedostossa muokkaamalla, saavutettaisiin haluttu ulkoasu. Elementtejä olivat wrapper, header, työnaytteet, navigation, ja footer. Näistä kaikki muut, paitsi *työnaytteet*, ovatkin elementtejä, joita lähes kaikki tämän päivän verkkosivut käyttävät pohjanaan. Kuvassa 10 nähdään Minnan portfolion rakenteen muutokset kahdessa *breakpointissa*.



**KUVA 10.** Minnan portfolion rakenne alle ja yli 720pikselin levyisille näytöille

Tyylitiedoston pohjaksi (joka linkitetään HTML-pohjaan *head* tagin sisään <link href="css/style.css" rel="stylesheet" />), latsin netistä tiedoston (<http://fluidbaselinegrid.com/>), jossa oli määriteltynä joitakin bodyn, otsikoiden ja typografian tyylejä, sekä ennen kaikkea yleisimmät *breakpointit* mediakyselyinä. Näiden mediakyselyiden sisälle voitaisiin myöhemmin laittaa ehtoja, jotka määrittävät mitä tapahtuu, kun päätelaite on tietyn kokoinen.

Oletusleveytenä portfolioissa toimisi kuitenkin 960px (standardi), jonka tyylimäärittelyt toimisivat pohjana, joita muokataan tarpeen tullen eri breakpointeissa. Kuvassa 11 nähdään kuinka breakpointeilla on tehty muutoksia headeriin, sekä taustakuvaan. Kun käyttäjä tarkastelee sivuja laajakuvana näytöllä, hän näkee reunoilla retrohenkiset palkit. Näytön koon pienentyessä *portfolio* – otsikko pienenee suhteessa näytön kokoon, sekä Minnan esittelyteksti rivittää itseään paremmin näytölle sopivaksi. Kun näytön koko on alle 720pikseliä, ei portfolio otsikkoa näytetä enää ollenkaan.



## KUVA 11. Breakpointien vaikutus käytännössä

Fontin koon, ja rivityksen sopivaksi näytölle skaalaaminen responsiivisesti tapahtuu käytännössä käyttämällä tyyleissä suhteellisia arvoja kuten *em* ja *%*. Oletus fonttikoko on usein selaimissa 16px, ja se merkataan *em* arvona 1. Alapuolella olevan esimerkin fonttikoko 5em, vastaisi siis fonttikokoa 80px. Kun tekstiä ei enää haluta näyttää ollenkaan, käytetään *display: none*; arvoa. Tekstin sopivan pystysuuntaisen rivityksen tarjoaminen on todella tärkeää, sillä on todettu, ettei käyttäjät mielellään vieritä sivua vaakasuunnassa.

```
.otsikko {  
  display: inline;  
  margin: 0 10 0 20;  
  padding-left: 2%;  
  font-size: 5em;  
  font-family: Futura, "Century Gothic", AppleGothic, sans-serif;  
  color: #222;  
  text-shadow: 1px 1px 1px rgba(0,0,0,.10);  
  float: left;  
}  
  
@media only screen and (max-width: 720px) {  
  .otsikko {  
    display: none;  
  }  
}
```

## KUVA 12. Esimerkki em ja % - arvojen käytöstä

Työnäytteitä div elementille annetaan prosenttiarvo, tässä tapauksessa 29,5 % riippuen siitä montako niitä halutaan näyttää vierekkäin. Tämä takaa sen, että nekin mukautuu näytön koon mukaan. Divin sisältämän kuvien alkuperäisellä koolla ei ole toimivuuden kannalta muuta merkitystä, kuin että niiden täytyy kaikkien olla samankokoisia. Kuvissa käytetään *max-width* : 100%: arvoa ilmoittamaan, ettei se voi koskaan olla isompi kuin elementtinsä. Näillä muutamalla rivillä koodia voidaan kattaa kaikki yli 960px levyiset näytöt, johon asetetaan breakpointi jolloin kuvat siirtyvät allekkain ja niille asetetaan kiinteä maximi leveys arvo, tässä tapauksessa 96px. Tässä portfolioissa työnäyte kuville on laitettu myös varjostukset, ja kursorin mennessä niiden päälle ne skaalautuvat CSS3:n *scale* ominaisuudella. Nämä ominaisuudet ovat uusia CSS3:ssa, eivätkä kaikki selaimet tue vielä niitä. Tästä syystä niitä on käytetty vain visuaalisen ilmeen rikastamiseen, eikä niitä käytetä palvelun kannalta kriittisissä toiminnoissa. Useissa näissä CSS3:n uusissa ominaisuuksissa selaimet tarvitsevat oman merkintänsä kuvan 13 mukaisesti.

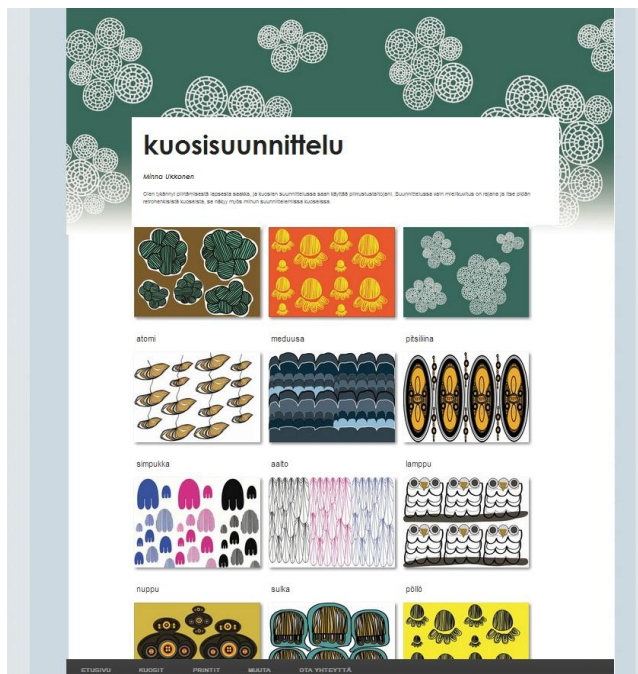
```

.tyonaytteet img {
-webkit-box-shadow: 4px 4px 10px rgba(0, 0, 0, 0.5);
-moz-box-shadow: 4px 4px 10px rgba(0, 0, 0, 0.5);
box-shadow: 4px 4px 10px rgba(0, 0, 0, 0.5);
}

.tyonaytteet img:hover {
-webkit-transform: scale(1.1) ;
-moz-transform: scale(1.1) );
-o-transform: scale(1.1) ;
transform: scale(1.1);
}

```

**KUVA 13. CSS3:n uusien ominaisuuksien merkintä**



**KUVA 14. Kuosisuunnittelun työnäytteet min-width: 1300px breakpointissa**



**Minna Ukkonen**

Olen tykännyt piirtämisestä lapsesta saakka, ja kuosien suunnittelussa saan käyttää piirustustaitojani. Suunnittelussa vain mielikuvitus on rajana ja itse pidän retrohenkisistä kuoseista, se näkyy myös minun suunnittelemissa kuoseissa.



atomi



meduusa



pitsiliina



simpukka



aalto



lamppu



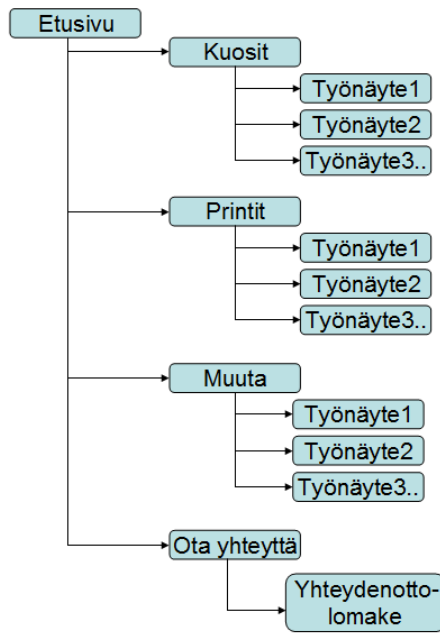
nappu



sulka

**KUVA 15. Kuosisuunnittelun työnäytteet max-width: 720px breakpointissa**

Navigoinnin selkeys on myös erityisen tärkeää, jotta käyttäjäystävällisyys säilytetään erilaisilla päätelaitteilla. On tärkeää, että käyttäjän ei tarvitse tehdä vaakasuuntaista vieritystä ja että valikon painikkeet ovat tarpeeksi suuret myös pienillä näytöillä. Navigoinnin rakenteen tulee myös olla helposti opittavissa.



**KUVA 16. Minnan portfolion navigaation rakenne**

Tässä portfolioissa isoilla näytöillä valikkopalkin paikka on asetettu kiinteäksi sivun alareunaan ja se seuraa mukana pystysuunnassa vieritettäessä. Valikoiden marginaali mukautuu tarpeen mukaan suuremmaksi tai pienemmäksi, jotta siisti ulkoasu säilytetään näytön koosta riippumatta. 720pikseliin on asetettu breakpoint, jossa valikot siirtyvät allekkain. Valikon paikka muuttuu myös staattiseksi, eli se ei enää seuraa käyttäjän pystysuuntaista vieritystä. Kuvassa 17 nähdään navigaation responsiivisuutta tyyli-tiedostossa.

```

footer nav.fixed {
  position: fixed;
  z-index: 1;
}

footer nav {
  width: 100%;
  max-width: 1440px;
  min-width: 760px;
  bottom: 0;
}

}

@media only screen and (max-width: 720px) {
  footer nav {
    width: 100%;
    min-width: 320px;
    min-height: 300px;
  }

  footer nav.fixed {
    position: static;
  }
}

```

### KUVA 17. Navigaation responsiivisuutta.

Sivun alareunassa oleva *footeri* on jaettu kahteen palstaan, joista toinen sisältää yhteystiedot, ja toinen sosiaalisen median linkit. Yli 720pikselin levyisillä näytöillä molemmat palstat ja niiden sisältämät kuvat ja tekstit kelluvat vierekkäin. Tästä pienemmillä näytöillä sosiaalisen median linkit siirtyvät yhteystietojen alle ja tässä tapauksessa kun niitä on kolme, jokaiselle varataan 30 % tilasta, jotta ne mukautuvat näytön koon mukaan.

```

@media only screen and (max-width: 720px) {
  footer .column {
    width: 100%;
    padding: 0;
    float: none;
  }
  footer .column.social {
    width: 30%;
    float: left;
    border-bottom: none;
  }
}

```

### KUVA 18. Footerin mukautuminen 720pikselin breakpointissa.

### 4.3 Käytännön toteutus (portfolio 2)

Kuten Minnankin portfolion, myös omani toteutuksen aloitin HTML rakenteesta. Tällä kertaa rakenne sisälsi elementit *page*, *inner*, *mast*, *intro*, *navigointi*, *main* ja *footer*. Portfolion rakenteen suunnitelma, eli käytännössä layouttien sommittelu, on hyvä tehdä kaikille merkittävälle breakpointeille.

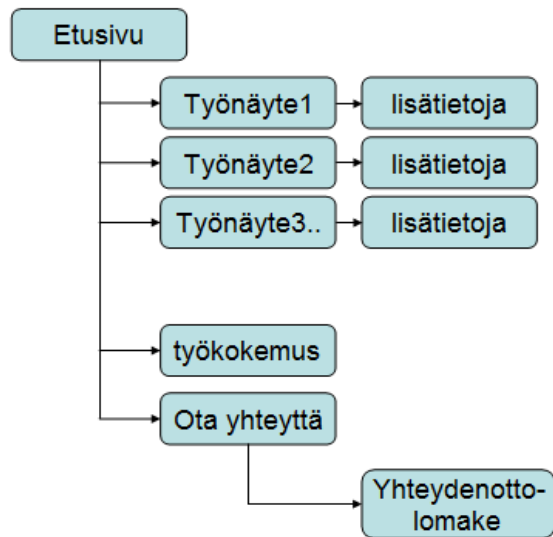


**KUVA 19. Oman portfolion rakenne (400px, 960px, ja 1300px breakpointeissa)**

Tärkeää responsiivisuuden ja mobile first - ajattelun kannalta oli, ettei sivulle tässä tapauksessa ahdeta ylimääräistä tietoa. Etusivu sisältää pienen esittelytekstin minusta, mutta muuten työnäytteet saavat puhua puolestaan. Muutenkin tyyli haluttiin pitää selkeänä ja navigointi helposti opittavana. Opittavuuden kannalta on tärkeää että navigaation rakenne pysyy samana kaiken kokoisilla näyttölaitteilla, vaikka navigaa-



tiopalkin sijainti vaihteleeekin päätelaitteesta riippuen. Kuvassa 20 nähdään minun portfolion navigaation rakenne.



**KUVA 20. Navigaation rakenne**

Tämä portfolio sisältää neljä merkittävää breakpointia, joissa kuvat, fontit, valikot ja muu sommittelu sopeutuu näytölle sopivammaksi. Rakenteen muutoksissa pyrittiin käyttämään teoriaosuudessa esiteltyä, Ethan Macrotetten (Macrotte 2010) kehittelemää  $\text{target/context} = \text{result}$  kaavaa. Muun muassa 1300pikselin breakpointissa valikko siirtyy kellumaan sivun oikeaan yläreunaan, valikolle halutaan varata 633pikseliä 960:stä, joten käyttämällä  $\text{target/context} = \text{result}$  kaavaa saadaan responsiiviseksi arvoksi 65.9375 prosenttia tilan leveydestä. Lisäksi valikossa on kolme painiketta, joista jokaiselle varataan kolmasosa tilasta miinus marginaalit, tulokseksi saadaan yhden painikkeen leveydeksi 31.121642969984202211 %. Kaavaa käyttäessä on tärkeää ottaa huomioon, mikä arvo asetetaan contextin arvoksi. Esimerkiksi tässä tapauksessa koko valikon arvo laskettiin standardi tilan leveydestä, eli 960pikselistä, ja jokaisen yksittäisen painikkeen contexti oli koko valikon leveys eli 633pikseliä. Tyylien viereen on suotavaa kommentoida arvot, joilla kaava on laskettu. Kuvassa 21 nähdään merkinnät käytännössä.

```

@media (min-width: 1300px) {
  .mast {
    float: none;
    width: auto;
  }
  ul.nav {
    float: right;
    margin: 40px 0 1em;
    text-align: center;
    width: 65.9375%; /* 633px / 960px */
  }
  ul.nav li {
    float: left;
    margin-right: 3.317535545023696682%; /* 21px / 633px */
    width: 31.121642969984202211%; /* 197px / 633px */
  }
}

```

### KUVA 21. Valikko min-width: 1300px breakpointissa

Samoja valikon painikkeiden arvoja voidaan käyttää myös max-width 600px breakpointissa. Koko valikon leveydeksi asetetaan 100% ja position: absoluteksi, koska pienemmillä näytöillä ei haluta sivun vasempaan reunaan muuta sisältöä.

```

@media (max-width: 600px) {
  ul.nav {
    border-top: none;
    margin: 0 auto;
    position: absolute;
    top: 0;
    width: 100%;
  }
}

```

### KUVA 22. Valikko max-width: 600px breakpointissa



**KUVA 23. Etusivun sommittelu erilaisilla näytöillä**

Työnäytteitä halutaan etusivun perusnäkyssä olevan kolme vierekkäin, joten jokaiselle varataan noin 31 % *main div* elementistä. Reunimmaisten oikea marginaali asetetaan nollassi, jotta kolmas työnäyte mahtuu muiden rinnalle. Suurilla näytöillä työnäytteitä halutaan näyttää 6 rinnakkain, joten jälleen käytetään  $\text{context/target} = \text{result}$  kaavaa hyökdyksi, laskettaessa kullekin työnäytteelle varattu tila ja niiden marginaalit. Nämä arvot tarvitsee merkata vain perusnäkyä reunimmaisille työnäytteille, muut työnäytteet mukautuvat niiden mukaan. Pienille päätelaitteille kuvat asetetaan allekkain kuten Minnankin portfolioissa, tällöin reunimmaisesta työnäytteestä ei tarvitse erillistä huomiota. Kuvassa 24 nähdään työnäytteiden tyylejä käytännössä.

```

.tyonaytteet {
float: left;
font-size: 10px;
line-height: 1.1;
margin: 0 3.317535545023696682% 1.5em 0;
text-align: center;
width: 31.121642969984202211%;
text-transform: uppercase;
letter-spacing: 0.05em;
}

li#portfolio,
li#javaee,
li#rumpusetti {
margin-right: 0;
}

@media (min-width: 1300px) {
...
.tyonaytteet,
li#portfolio,
li#javaee {
margin-right: 3.317535545023696682%;          /* 21px / 633px */
width: 13.902053712480252764%;             /* 88px / 633px */
}
}

@media (max-width: 400px) {
...
.tyonaytteet{
background: #ECE5B6;
border: none;
border-bottom: 1px solid #ccc;
padding: 0;
width: 100%;
float: none;
margin: 0 0 2% 0;
overflow: auto;
clear: both;
}

.tyonaytteet img {
width: 100%;
max-width: 96px;
float: left;
margin: 6px;
}
}

```

## KUVA 24. Työnäytteiden responsiivisuus etusivulla

Etusivun työnäytteitä klikkaamalla käyttäjä pääsee lukemaan lisätietoja yksittäisestä työnäytteestä. Myös tällä sivulla on huomioitu responsiivisuus merkkamalla paddingit, marginaalit, fontit ja rakenteet suhteellisilla arvoilla *em* tai *%*. Infoteksti rivittyi miellyttävästi luettavaksi, ja kun näyttölaite on tarpeeksi laaja, kuva ja teksti siirtyvät rinnakkain.

```

#lisainfo div.featureImage {float:left;width:48.5%; }
#lisainfo div.featureImage a {display: block; margin:0;padding:0;border:
#lisainfo p.featureImage a img {display: block;}

#lisainfo .featureText {float:right; width:48.5%;}
#lisainfo .featureText h2 {padding:0 1em 0 0;
margin:.06667em 0 0 0;font-size: 1.875em;color: #333;line-height: 1.4;

@media (max-width: 400px) {
#lisainfo {padding:.625em .625em 1em .625em;margin-top:1em;}
#lisainfo .featureText h2 {font-size: 1.5em;padding:0;margin:.3em 0 0 0
#lisainfo .entry-meta{display:none;}
#lisainfo div.featureImage {float:none;width:100%;}
#lisainfo .featureText {float:none; width:auto;}
}

@media (max-width:1300px) {
#lisainfo div.featureImage {float:none;width:100%;}
#lisainfo .featureText {float:none; width:auto;}
}

```

**KUVA 25.** Työnäyte sivujen kolme merkittävää breakpointia ja suhteellisten arvojen käyttöä



**KUVA 26.** Esimerkki työnäyte sivun responsiivisuudesta. Kuvassa kännykkä-, tabletti-, ja laajakuvaversiot

#### 4.4 Portfolioiden testaus

Molempien portfolioiden käytettävyyttä testattiin työpöytäkoneella yleisimmillä selaimilla, eli Internet Explorerilla, Mozilla Firefoxilla ja Google Chromella. Internet Explorerin kanssa ilmeni eniten selainkohtaisia ongelmia, muun muassa HTML5 lomakkeiden, sekä satunnaisten kuvien kanssa. Havaitut ongelmat saatiin kuitenkin työpöytäkoneilta korjattua. Mobiili,- sekä tabletilayoutteja testattiin HTC Desire HD, Samsung Galaxy S3, iPhone 4 älypuhelimilla, sekä Samsung galaxy 10 tabletilla. Havaittiin, että Minnan portfolio sisältää paljon korkealaatuisia kuvia, joten sivun latausajat nousevat melko suureksi. Minnan portfolioon olisi voinut myös laittaa jonkinlaisen *menu* painikkeen esimerkiksi sivun yläkulmaan, jolla pääsisi nopeasti valikoihin, ilman ylimääräistä pystysuuntaista vieritystä. Nämä ongelmat tullaan korjaamaan myöhemmin. Lisäksi sivut sisälsivät alun perin jQueryn lightboxin, jolla kuvat aukevat uuteen ikkunaan suurempina. Tämä ominaisuus kuitenkin poistettiin testausvaiheessa, kun todettiin sen responsiivisuuteen liittyvät ongelmat. Omassa portfolioissani ei mobiililaitteilla suurempia ongelmia havaittu, latausajat olivat hyviä ja yhteydenotomake toimi kiitettävästi. Muutamassa työnäytteessä otsikkoteksti pursusi divin reunan yli tietyn kokoisella näytöllä.

### 5 PÄÄTÄNTÖ

Johtopäätöksenä voidaan todeta, että kun verkkosivua aletaan toteuttaa puhtaalta pöydältä, responsiivisuus saavutetaan suhteellisen helposti. Tarvitsee vain muistaa joustava sommittelu, joustavat kuvat, sekä mediakyselyt. Toisaalta, jo olemassa olevan verkkosivun muuttaminen responsiiviseksi on huomattavasti haastavampaa, kun etenkin joustavaa sommittelu ei välttämättä voida toteuttaa.

Kun suunnitellaan responsiiviset verkkosivut, säästetään niin toteutus-, kuin ylläpito-kustannuksissakin, verrattuna erillisen työpöytäversion, kännykkäversion ja vaikkapa tablettiversion tekemiseen. Tästä syystä kaikkien verkkopalveluja suunnittelevien tahojen tulisi todella miettiä, löytyykö perusteluja erillisten mobiiliversioiden tekemiselle, jos päädytään siihen tulokseen, ettei responsiivista verkkosuunnittelua käytetä hyödyksi.

Kuvien merkitys käyttäjäystävällisyyden kannalta nousi esiin Minnan portfolioissa. Testausvaiheessa todettiin, että hienot responsiiviset verkkosivut eivät yksin riitä. Latausaikojen ollessa liian korkeita, käyttäjä ei todennäköisesti jää odottamaan.

CSS3:n ja HTML5:n uusia ominaisuuksia voidaan huoletta käyttää rikastamaan käyttäjäkokemusta, mutta palvelun kannalta kriittisissä toiminnoissa niitä kannattaa tois-  
taiseksi välttää. Kannattaa kuitenkin muistaa, että CSS3 ominaisuudet tarvitsevat usein selainkohtaisen etuliitteen.

Yleisesti ottaen, jos et ole aivan sataprosenttisen varma, millä laitteella julkaisuasi tullaan selaamaan, kannattaa suosia responsiivista verkkosuunnittelua. Näin voidaan kattaa suuri laitteiden kirjo. Toteutus ei vie juurikaan enempää aikaa kuin perinteisen verkkopalvelun toteutus, joten negatiivisia puolia on vaikea löytää.

Jatkotutkimuksena voisi selvittää JavaScriptin, esimerkiksi jQuery-, Ajax- ja Flash-tekniikoiden soveltamisen responsiivisilla sivuilla. Myös julkaisujärjestelmällä (CMS) toteutetun sivuston muokkaaminen responsiiviseksi on kiinnostava jatkoselvityksen aihe.

## LÄHTEET

Ahonen, Kari. 2011. Oikea lähestymistapa. WWW-dokumentti. .  
<http://www.webwellhot.fi/oikea%20lahestymistapa.html>. Päivitetty 13.05.2011. Luettu 2.12.2012.

Berners-Lee, Tim. 2010. WWW-dokumentti.  
<http://www.scientificamerican.com/article.cfm?id=long-live-the-web>. Päivitetty 22.11.2010. Luettu 16.12.2012.

Bradley, Steven. 2011. How to create flexible images and media in CSS layouts. WWW-dokumentti. <http://www.vanseodesign.com/css/flexible-images/>. Päivitetty 23.06.2011. Luettu 21.12.2012.

CSS Units. 2012. WWW-dokumentti. [http://www.w3schools.com/cssref/css\\_units.asp](http://www.w3schools.com/cssref/css_units.asp). Ei päivitystietoja. Luettu 20.12.2012.

Jaakkola, Tapio, 2011. Arvio HTML5:n uusien ominaisuuksien sovellettavuudesta. PDF-dokumentti.  
[http://www.gofore.com/sites/default/files/Jaakkola\\_Tapio\\_Diplomityo.pdf](http://www.gofore.com/sites/default/files/Jaakkola_Tapio_Diplomityo.pdf). Luettu 20.12.2012.

Julkaisujärjestelmät. 2011. WWW-dokumentti.  
<http://www.tulos.fi/julkaisujarjestelmat/>. Ei päivitystietoja. Luettu 07.12.2012

Julkaisujärjestelmät. 2012. WWW-dokumentti. <http://www.ubinet.fi/www-suunnittelu/julkaisuj%C3%A4rjestelm%C3%A4>. Ei päivitystietoja. Luettu 17.12.2012.

Kahdeksan yleistä virhettä. 2010. WWW-dokumentti.  
[http://www.wteche.eu/fi/modules/web\\_design/8\\_common\\_mistakes/](http://www.wteche.eu/fi/modules/web_design/8_common_mistakes/). Ei päivitystietoja. Luettu 1.12.2012.

Keith, Jeremy. 2010. HTML5 for webdesigners, A book Apart. New York: Jeffrey Zeldman.



Keith, Jeremy. 2010. WWW-dokumentti. <http://adactio.com/journal/1716/>. Päivitetty 9.12.2010. Luettu 16.12.2012.

Larvanko, Lasse. 2007. WWW-dokumentti. <http://www.inventive.fi/web-ei-ole-visuaalinen-media>. Ei päivitystietoja. Luettu 9.12.2012.

Macrotte, Ethan. 2010. Responsive web design, A book Apart. New York: Jeffrey Zeldman.

Mitä on verkkosuunnittelu?. 2010. WWW-dokumentti. [http://www.wteche.eu/fi/modules/web\\_design/what\\_is\\_web\\_design](http://www.wteche.eu/fi/modules/web_design/what_is_web_design). Ei päivitystietoja. Luettu 1.12.2012.

Mudge, JT. 2012. WWW-dokumentti. <http://sixrevisions.com/mobile/native-app-vs-mobile-web-app-comparison/>. Päivitetty 11.07.2012. Luettu 17.12.2012.

Page strugture and site design. 2009. WWW-dokumentti. <http://webstyleguide.com/wsg3/6-page-structure/3-site-design.html>. Ei päivitystietoja. Luettu 11.12.2012.

Rieger, Stephanie. 2011. WWW-dokumentti. <http://www.slideshare.net/yiibu/pragmatic-responsive-design>. Päivitetty 12.12.2011. Luettu 14.12.2012.

Scaglione, Jenna. 2012. WWW-dokumentti. <http://www.webdesignerdepot.com/2012/07/insider-views-on-responsive-design-the-debate-continues/>. Päivitetty 02.07.2012. Luettu 18.12.2012.

Söderholm, Maria. 2003. Esteettömät verkkosivut. WWW-dokumentti. <http://appro.mit.jyu.fi/essikurssi/ika/t1/>. Päivitetty 3.011.2003. Luettu 13.12.2012.

Taali, Marko. 2010. Web is dead, kuka voittaa ja miten?. WWW-dokumentti. <http://dahmon.org/taali/tags/verkkosuunnittelu/>. Ei päivitystietoja. Luettu 3.12.2012.

The CSS saga. 2012. WWW-dokumentti. <http://www.w3.org/Style/LieBos2e/history/>.  
Ei päivitystietoja. Luettu 18.12.2012.

Using site speed in web search ranking. 2010. WWW-dokumentti.  
<http://googlewebmastercentral.blogspot.fi/2010/04/using-site-speed-in-web-search-ranking.html>. Päivitetty 10.04.2010. Luettu 4.12.2012.

Viitanen, Maria. 2007. PDF-dokumentti.  
<http://users.tkk.fi/~mfviitan/Studio4/vuorovaikutusessee.pdf>. Päivitetty 4.04.2007.  
Luettu 13.12.2012.

Wireless flexible keyboard. 2012. WWW-dokumentti.  
<http://jiaxingwear.en.made-in-china.com/productimage/wbHQvFpxrZVL-2f0j00EseQJWYgYtoK/China-Wireless-Flexible-Keyboard.html>. Ei päivitystietoja.  
Luettu 20.12.2012.

Wroblewski, Luke. 2011. Mobile first, A book Apart. New York: Jeffrey Zeldman.

Zeldman, Jeffrey. 2010. CSS3 for webdesigners, A book Apart. New York: Jeffrey Zeldman,

1BlizzAdmin. 2008. The history and future of CSS. WWW-dokumentti.  
<http://newsletter.blizzardinternet.com/the-history-future-of-css/2008/04/11/>. Päivitetty 11.04.2008. Luettu 18.12.2012.