

Joni Kiviaho

**OAMK:N TURVALLISUUSOHJEIDEN KEHITTÄMINEN RESPONSIIVISEKSI WEB-  
SIVUSTOKSI**

# **OAMK:N TURVALLISUUSOHJEIDEN KEHITTÄMINEN RESPONSIIVISEKSI WEB-SIVUSTOKSI**

Joni Kiviaho  
Opinnäytetyö  
Kevät 2015  
Tietojenkäsittelyn koulutusohjelma  
Oulun ammattikorkeakoulu

## TIIVISTELMÄ

Oulun ammattikorkeakoulu  
Tietojenkäsittelyn koulutusohjelma  
Web-sovelluskehitys

---

Tekijä(t): Joni Kiviaho

Opinnäytetyön nimi: Oamk:n turvallisuusohjeiden kehittäminen responsiiviseksi web-sivustoksi

Työn ohjaaja: Ritva Virkkala

Työn valmistumislukukausi- ja vuosi: Kevät 2015

Sivumäärä: 40

---

Tutustuin opinnäytetyössäni responsiivisen web-suunnittelun historiaan ja responsiivisiin tekniikoihin. Esittelen myös varsinaisen opinnäytetyöni eli sivuston, jonka suunnittelin ja rakensin responsiivisena Oulun ammattikorkeakoulun käyttöön. Sivuston tarkoituksena on palvella mobiilikäyttäjiä mahdollisissa hätätilanteissa ja koota koulun turvallisuusohjeet yhteen helposti löydettävään paikkaan. Tähän asti turvallisuusohjeet ovat olleet hajallaan intranetin uumenissa, joten hätätilanteessa niistä ei juurikaan olisi ollut apua. Tavoitteena on siis tarjota helposti tavoitettava, selkeät toimintaohjeet sisältävä tietokanava, joka parantaa sekä opiskelijoiden että koulun henkilökunnan turvallisuutta ja valmiuksia toimia erilaisissa hätätilanteissa. Toimeksiantajana toimivat turvallisuusvastaavat Sami Niemelä ja Pekka Ojala.

Opinnäytetyönä tehdyn sivuston kriteereinä olivat responsiivisuus, mobiiliystävällisyys, selkeys ja keveys, tarkoittaen latautumisenopeuden huomioon ottamista esimerkiksi hitaammilla mobiiliverkkoyhteyksillä. Responsiivisuuden toteuttamisessa ei ole käytetty asettelukehyksiä, kuten Bootstrapia, sillä ne ovat liian raskaita ja monipuolisia näinkin pienelle sivustolle. Sen sijaan sivustolla käytetään ainoastaan CSS-kielen mediakyselyjä, joilla mukautetaan sivuston ulkoasua tiettyjen laitteiden tai selainleveyksien välillä. Sivusto on rakennettu käyttämällä HTML-, CSS-, JavaScript- ja PHP-ohjelmointikieliä. Sivuston testaus suoritettiin laitetestauksena.

Responsiivisuuteen liittyvää suomenkielistä kirjallisuutta tai muita lähteitä ei juurikaan ole saatavilla, joten turvauduin opinnäytetyössäni englanninkielisiin verkkoartikkeleihin sekä muihin alan sivustoihin. Alan nopea kehitys asettaa haasteita kirjallisille julkaisuille, sillä nykypäivänä tieto vanhenee nopeasti. Onnistuinkin löytämään sekä tuoretta tutkimustietoa että artikkeleja, jotka antoivat ajattelemisen aihetta ja uusia näkökulmia alalla työskentelyyn. Esimerkiksi mobile first -ajattelutapa on asia, jota aion jatkossa ehdottomasti hyödyntää omassa työssäni, ja olen yleisesti ottaen paljon tietoisempi mobiilikäyttäjien osuudesta internetliikenteessä. Mobiilikäyttäjien ja samalla -laitteiden määrä tulee jatkossakin kasvamaan ja monipuolistumaan, joten on täysin perusteltua ottaa tämä suunnatonta vauhtia kasvava käyttäjäryhmä entistä paremmin huomioon. Voin todeta aihetta tutkittuani, että responsiivisuus on vain osa mobiiliystävällistä web-suunnittelua, eikä internetin käyttö ole enää samanlaista kuin vaikkapa viisi vuotta sitten. Kehityksen mukana pysyminen ei vaadi paljon, ja se kannattaa.

---

Asiasanat: Web-suunnittelu, web-ohjelmointi, responsiivisuus, mobiiliystävällisyys, mediakysely

## ABSTRACT

Oulu University of Applied Sciences  
Degree Programme in Business Information Systems  
Web Application Development

---

Author(s): Joni Kiviaho

Title of Bachelor's thesis: Development of Oamk security instructions into a responsive website

Supervisor(s): Ritva Virkkala

Term and year of completion: Spring 2015

Number of pages: 40

---

For my bachelor's thesis I explored the history of responsive web design and responsive techniques. I also present the practical aspect of my thesis, which is a responsive website that I designed and built for Oulu University of Applied Sciences. The objective of the site is to provide an easy access to the school's security instructions for mobile users in case of emergency. Thus far the safety instructions had been buried in the depths of the school's intranet, which means they would have been of little use were they really needed. Therefore, the aim was to create a safety information channel with easy access, which would improve the safety and aiding capabilities of both the students and the staff in different emergency scenarios. The project was assigned by security officers Sami Niemelä and Pekka Ojala.

The criteria for the website were responsiveness, mobile friendliness, easy usability and lightness, meaning taking the loading speed into account when using slower mobile networks. Responsiveness was utilized with only CSS media queries instead of frameworks such as Bootstrap, because they would have been too heavy and complicated for such a small site. The site was built with HTML, CSS, JavaScript and PHP. Testing was accomplished via actual device testing.

Finnish literature or other sources regarding responsive web design are hard to find, which led me to use English web articles and industry websites instead. The field of web design is growing rapidly, which makes it challenging for physical publications to deliver up-to-date information. However, I managed to find both recent studies and articles which gave me something to think about and brought new aspects to working in web design. For instance, mobile first is something I am definitely going to use in my own work, and I certainly know to focus more on mobile Internet users. The amount of mobile devices and users will continue to grow, so it is justified to take this ever-growing segment of consumers better into account. After studying the subject, I can honestly say that the Internet is not the same as it was some five years ago, but it does not take much to keep up and it most definitely is worth it.

---

Keywords: Web design, programming, responsiveness, mobile friendliness, media query

# SISÄLLYS

1	JOHDANTO .....	6
2	MUKAUTUVA WEB .....	7
2.1	Mobiililaitteiden ja responsiivisuuden historiaa .....	7
2.2	Responsiiviset web-tekniikat .....	10
2.2.1	CSS (Cascading Style Sheet) .....	11
2.2.2	Media queryt .....	12
2.2.3	Frameworkit .....	13
2.2.4	Mobile first.....	15
2.2.5	Responsiivisen web-suunnittelun kritiikkiä .....	17
3	OAMK:N TURVALLISUUSOHJEET -SIVUSTON TOTEUTUS.....	21
3.1	Suunnittelu .....	21
3.2	Sivuston vaatimukset ja rakenne .....	23
3.3	Ulkoasu ja käytettävyys .....	24
3.4	Latausnopeuden optimointi .....	26
3.5	Navigointi.....	26
3.6	Responsiivisuus .....	27
3.7	Testaus ja koodin validointi .....	27
4	JOHTOPÄÄTÖKSET .....	29
5	POHDINTA .....	36
6	LÄHTEET .....	38

# 1 JOHDANTO

Älypuhelinien, tablettien ja muiden älylaitteiden käyttö lisääntyy jatkuvasti, ja uusia laitemalleja ilmestyy markkinoille tämän tästä – pelkästään jotkut valmistajat julkaisevat uuden mallin jopa puolen vuoden välein. Tämä älylaitteiden koko ajan päivittyvä kirjo on vaikuttanut ihmisten internetikäyttäytymiseen ratkaisevasti, ja web-suunnittelijoiden on ollut pakko reagoida tähän muutokseen muuttamalla sisällön esitystapaa joustavammaksi. Vuosia sitten riitti, että sivusto oli käyttökelpoinen pöytätietokoneilla koska vaihtoehtoja ei juuri ollut, mutta viime vuosikymmenen aikana web-sivustojen vaatimukset ovat muuttuneet radikaalisti nimenomaan älylaitteiden yleistymisen myötä, ja joustamattomuus on peruuttamattomasti pois muodista. Tämä opinnäytetyö paneutuu juuri tähän aiheeseen ja pyrkii valottamaan esimerkkien myötä, mitä responsiivisuus tarkoittaa ja miksi se on tärkeää.

Perehdyn opinnäytetyössäni responsiivisten web-tekniikoiden ja mobiililaitteiden historiaan ja kehitykseen, sekä erilaisiin tapoihin toteuttaa responsiivisuutta. Esittelen myös CSS-kielen peruseräkkeet ja sen mahdollistamat mediakyselyt, jotka ovat yleisin tapa luoda mukautuvia ulkoasuja sivustoille. Tutkin myös, miksi jotkut suhtautuvat responsiiviseen web-suunnitteluun kielteisesti ja miksi se ei heidän mielestään maksa aikaa ja vaivaa. Näiden näkökulmien ja aiheeseen perehtymisen kautta toivon löytäväni tapoja, joilla voin parantaa omaa työskentelyäni ja kehittää ammattitaitoani.

Lopuksi esittelen opinnäytetyöni käytännön osuuden eli responsiivisen sivuston, jonka suunnittelin ja ohjelmin Oulun ammattikorkeakoulun käyttöön. Sivuston tarkoituksena on koota Oamk:n turvallisuusohjeet paikkaan, josta voi nopeasti tarkistaa, kuinka erilaisissa hätätilanteissa tulee toimia. Aikaisemmin turvallisuusohjeet ovat sijainneet milloin missäkin, ja mikäpä sen parempi ja nykyaikaisempi ratkaisu kuin mobiiliystävällinen sivusto. On kuitenkin huomautettava, ettei kyseessä ole valmis kokonaisuus vaan pikemminkin demo: sivusto toimii lähinnä havainnollistavana esimerkkinä responsiivisuudesta ja on jatkokehitettävissä esimerkiksi toisessa opinnäytetyössä tai jollakin opintojaksolla. Hätätilanteistakin sivustolle on valikoitu vain kourallinen. Joka tapauksessa toivon, että tämä opinnäytetyö konkretisoi responsiivisuuden merkityksen; tällä alalla se tulee hallita.

## 2 MUKAUTUVA WEB

Tässä kappaleessa kerron web-sivustojen responsiivisuudesta ja sen historiasta sekä eri tekniikoista, joilla responsiivisuutta voidaan toteuttaa.

Responsiivisuudella tarkoitetaan web-suunnittelusta puhuttaessa sitä, että sivustot rakennetaan sopeutumaan (skaalautumaan) käyttäjän laitteeseen, kuten esimerkiksi älypuhelimeen ja/tai tablettiin. Nykyään yhä useamman verkkoprojektin yksi vaatimus onkin juuri mobiilistävällisyys, mikä ei liene yllättävää, kun otetaan huomioon mobiililaitemarkkinoiden kehitys viime vuosina. (Leiniö 2012, viitattu 7.3.2015.) Yleensä responsiivisuus ymmärretään tarkoittamaan lähinnä kannettaville laitteille tarkoitettua web-sisällön mukautuvuutta, mitä se pääasiassa onkin, mutta laajemmassa skaalassa responsiivisuuden piiriin kuuluvat periaatteessa kaikki laitteet, joilla voi käyttää internetiä – kyse ei ole vain siitä, *millä* internetiä käytetään, vaan myös *miten*. Esimerkiksi kosketusnäytöt eivät enää ole vain älypuhelimien ja tablettien erikoisuus, vaan myös varta vasten valmistettuja pöytätietokoneiden näyttöjä voidaan käyttää koskettamalla. Responsiivisuudessa ei siis ole kyse ainoastaan sivuston skaalautuvuudesta näyttökoon mukaan, vaan myös siitä, että sivustoa voidaan *käyttää* laitteen koosta riippumatta (Six 2013, viitattu 7.3.2015).

### 2.1 Mobiililaitteiden ja responsiivisuuden historiaa

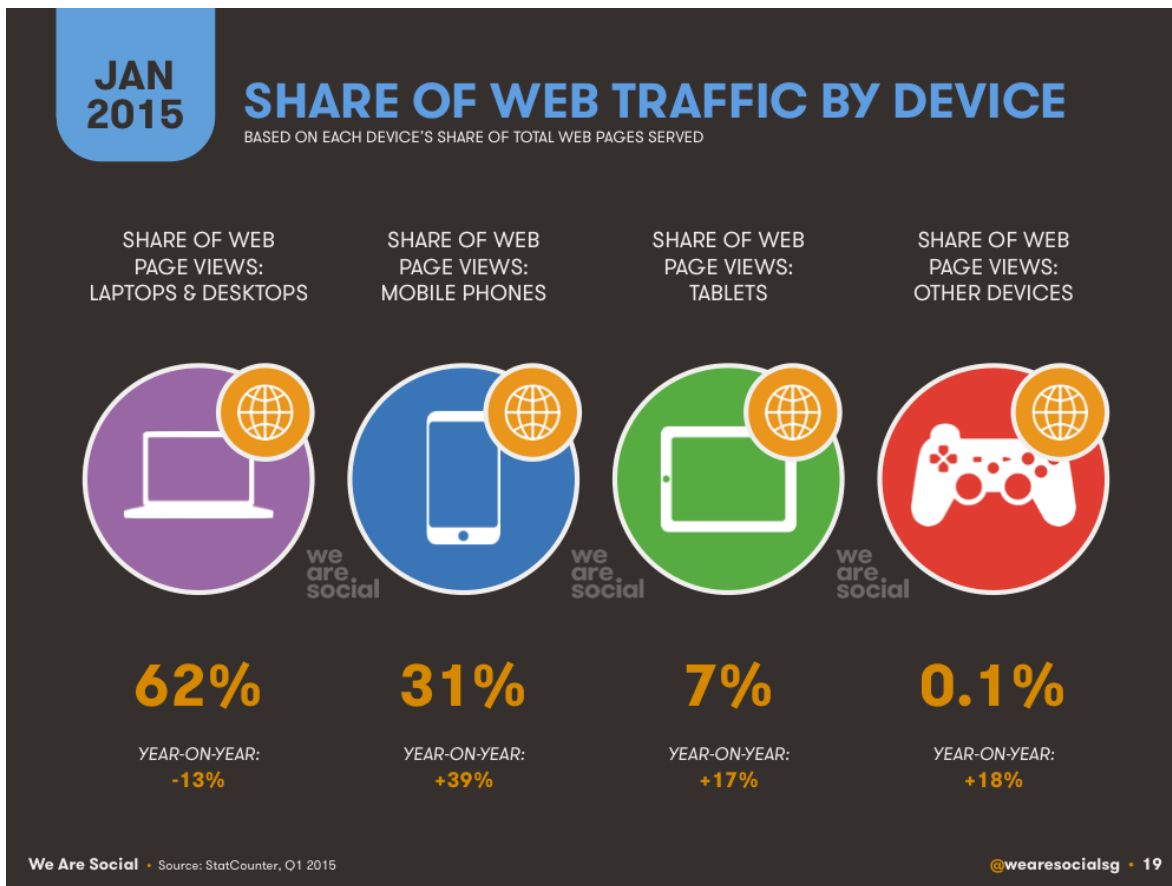
Mobiililaitteilla – lähinnä kännyköillä – on päässyt internetiin jo 1990-luvun loppupuoliskolla muun muassa WAP-protokollaa käyttäen, mutta sen ajan kännykät kykenivät suhteellisen pitkään näyttämään ainoastaan tekstipohjaista dataa. 2000-luvun lopulla älypuhelimien yleistyttyä mobiililaitteiden selaimet alkoivat jo olla teknisesti sitä luokkaa, että niiden onnistui näyttää myös ns. ”oikeita” web-sivuja. Siihen asti web-sivut oli rakennettu kiinteälevyisiksi (fixed width), tarkoittaen lähinnä siihen aikaan yleisimpiin (800 tai 1024 pikseliä leveisiin) näyttöihin sopiviksi, koska muun kokoisia näyttöjä ei liiemmin ollut. (Soojian 2015, viitattu 9.3.2015.)

Aikaisemmin web-sivustojen ”mukautuvuus” toteutettiin esimerkiksi siten, että käyttäjän selaimen nimi ja versio selvitettiin vaikkapa JavaScriptillä (browser sniffing, nuuskinta). 1990-luvun myöhemmällä

puoliskolla web-standardien implementointi ei ollut yhtenäistä eri selaimien välillä; esimerkiksi kehykset (frames) olivat yhdessä vaiheessa ominaisuus, jota kaikki selaimet eivät tukeneet, ja sivut tuli optimoida sen mukaan, onko käyttäjän selaimessa haluttuja ominaisuuksia vai ei. Nuuskinnasta saatujen tietojen perusteella käyttäjälle voitiin siis näyttää sivuston versio, joka oli kooditasolla optimoitu hänen selaimelleen. Nuuskinnan suurimmaksi ongelmaksi muodostui kuitenkin lopulta se, että vaikka jokaisella selaimella oli (ja on edelleenkin) periaatteessa yksilöllinen User Agent-merkkijonotunniste, tuotevalmistajat manipuloivat näitä UA-tunnisteita saadakseen sivustot näyttämään käyttäjilleen paras mahdollinen versio, ja selainten sekä niiden versioiden lisääntyessä nuuskintaskriptit eivät enää pysyneet perässä. UA-tunnisteiden luotettavuus siis kärsi, ja pikku hiljaa nuuskinnasta luovuttiin. (Web Education Community Group 2013, viitattu 7.3.2015.)

Tämän päivän mobiilitrendin katsotaan alkaneen vuonna 2007, kun yhdysvaltalainen Apple julkaisi ensimmäisen iPhonensa. Tätä ennen internetin käyttö – kuten me sen ymmärrämme – oli sidottu lähinnä pöytä- ja kannettaviin tietokoneisiin, mutta iPhonen tultua markkinoille mobiililaitteet ovat yleistyneet ja myös monipuolistuneet, ja internetiä voidaan tänä päivänä käyttää melkein missä ja millä tahansa. We Are Social -sivusto julkaisi äskettäin 376-sivuisen *Digital, Social and Mobile in 2015* -raportin, josta selviää muun muassa, että tammikuussa 2015 maailman kaikista web-sivujen latauksista 62% tapahtui pöytä- ja kannettavilla tietokoneilla ja loput 38% muilla laitteilla (kuvio 1). Samasta kuviosta selviää myös, että tietokoneiden osuus sivulatauksista laski edellisvuoteen nähden 13%, kun taas muiden laitteiden osuus nousi yhteensä 74%. (Kemp 2015, viitattu 7.3.2015.)





KUVIO 1. Eri laitteiden osuus internetliikenteestä tammikuussa 2015.

Responsiivisen ajattelun pioneerina pidetään web-suunnittelija Ethan Marcottea, joka avasi keskustelun aiheesta vuonna 2010 julkaisemassaan *Responsive Web Design* -artikkelissa. Siinä käsiteltiin muun muassa web-suunnittelun mullistumista mobiilikäyttäjien eksponentiaalisen kasvun johdosta ja esiteltiin ajatus, että tähän mullistukseen tulisi vastata joustavalla web-suunnittelulla. Marcotten mielestä tämä voitaisiin saavuttaa hyödyntämällä joustavia ruudukkoita (flexible/fluid grids) ja kuvia sekä CSS:n mediakyselyjä. On kuitenkin painotettava, ettei Marcotte sinänsä keksinyt mitään uutta, vaan yhdisti nämä jo olemassa olevat työkalut ja komponentit konseptiksi nimeltä Responsive Web Design. (Soojian 2015, viitattu 9.3.2015.)

RWD on tärkeä konsepti, koska mobiilikäyttäjien yleistyessä yritysten web-sivustojen tulee toimia saumattomasti kaikilla laitteilla, jotta ne voivat menestyä. Responsiivisiksi suunnitellut sivustot eivät tarvitse useita www-osoitteita, vaan ne sijaitsevat yhdessä paikassa ja hyödyntävät samaa koodia. Ideana on, että sivusto havaitsee käyttäjän laitteen ja sovittaa sisältönsä sen mukaan ilman, että

käyttäjän tarvitsee tehdä mitään – varsinkaan siirtyä täysin toiseen www-osoitteeseen. Tämä on paitsi kustannustehokasta myös helppo ylläpitää, kun samaa sisältöä ei tarvitse muuttaa monessa paikassa. (Soojian 2015, viitattu 9.3.2015.)

Alexandra Giarratano käsittelee artikkelissaan *Responsive Web Design Is Essential In 2015* (viitattu 3.5.2015) responsiivisen web-suunnittelun nykytilaa. Giarratano toteaa, että kun vielä muutama vuosi sitten oli hyväksyttävää esittää ainoastaan oleellinen tieto mobiilisivustolla, nykyään 85% internetin käyttäjistä on sitä mieltä, että yrityksen sivuston mobiiliversion tulisi olla yhtä hyvä tai jopa parempi kuin ns. työpöytäversio. Toisin sanoen kuluttajat odottavat samaa käyttökokemusta kaikilla laitteillaan, tarkoittaen muun muassa navigointia, hakupalkkia ja samojen toimintojen suorittamista (esim. ostosten tekeminen ja lomakkeen lähettäminen). Kuluttajilla on nykyään käytettävissään useita laitteita, ja koska he tulevat vierailemaan samalla sivustolla useammalla päätteellä, on turhauttavaa, jos sivuston toiminnot ovat laiteriippuvaisia. Kun käyttäjät eivät ole tyytyväisiä (mobiili)sivustoon ja poistuvat muutaman klikkauksen jälkeen, sivuston poistumisprosentti kasvaa. Google rankaisee korkeasta poistumisprosentista antamalla sivustolle matalamman arvosanan, mikä vaikuttaa sivuston liikenteeseen. Responsiivinen suunnittelu auttaa estämään tätä, koska silloin sivusto on mukautuva ja yhdenmukainen laitteella kuin laitteella, eivätkä käyttäjät turhaudu ja poistu.

Giarratanon artikkelissa mainitaan myös, että Google on äskettäin muuttanut hakukonettaan mobiiliystävällisemmäksi. Marraskuussa 2014 Google alkoi lisätä hakutuloksiinsa merkintöjä sivustojen mobiiliystävällisyydestä, ja huhtikuussa 2015 se ilmoitti alkaneensa parantaa mobiiliystävällisten sivustojen arvioita ja niiden sijoituksia hakutuloksissa. Vaikka sekä responsiiviset että mobile only -sivustot läpäisevät tämän seulan, Google suosittelee käyttämään responsiivista web-suunnittelua sen sijaan, että mobiilikäyttäjille tehtäisiin sivustosta täysin oma versionsa. (Giarratano 2015, viitattu 3.5.2015.)

## **2.2 Responsiiviset web-tekniikat**

Responsiiviset web-tekniikat koostuvat joustavista ruudukoista (flexible grids) sekä HTML5- ja CSS3-kielistä (Leiniö 2012, viitattu 7.3.2015). HTML-kuvauskielellä luodaan sivuston rakenne, ja CSS-tyylimäärittelyillä kerrotaan miltä sivuston tulee näyttää, niin suurella kuin pienelläkin näytöllä. On

olemassa myös valmiita asettelukehyksiä (framework), joilla voidaan nopeastikin luoda joustava ja skaalautuva web-sivusto.

### 2.2.1 CSS (Cascading Style Sheet)

CSS eli *Cascading Style Sheets* on tyylimäärittelykieli, jolla kerrotaan selaimelle kuinka web-sivu tulee esittää käyttäjälle. CSS sai alkunsa jo vuonna 1997, kun web-kehittäjät halusivat erottaa sisällön ulkoasusta. CSS:n kehitti World Wide Web Consortium (W3C), jotta HTML-koodin sekaan ei sekoitettaisi mitään tyylittelyä, vaan se määrittäisi ainoastaan dokumentin rakenteen, ja kaikki ulkoasulliset seikat puolestaan tyylitettäisiin CSS:llä erillisessä tiedostossa. (W3Schools 2015a, viitattu 8.3.2015). Tämä erittely alkaa olla jo standardi; CSS:ää käyttämättömien web-kehittäjien katsotaan jo olevan ajastaan jäljessä, ja tänä päivänä CSS:n ymmärtämistä pidetäänkin vähintään yhtä tärkeänä kuin HTML-osaamista (Kymin 2015, viitattu 7.3.2015).

Kuten HTML:stä, myös CSS:stä julkaistaan aika ajoin uusia versioita uusilla ominaisuuksilla; tämänhetkinen versio on 3, ja kaikki nykyaikaiset selaimet tukevat täysin CSS:n versiota 1, suurinta osaa versiosta 2 ja osittain versiota 3 (Kymin 2015, viitattu 7.3.2015). CSS-säännöt koostuvat selektorista (selector) ja määryksestä (declaration): selektori on tyyliteltävä objekti ja valitaan joko sen tyyppin, yksilöllisen tunnisteen (id) tai luokan (class) perusteella, ja määrykset sijoitetaan aaltosulkujen sisään puolipisteellä erotellen (W3Schools 2015b, viitattu 7.3.2015). Määrykset tapahtuvat ryhmissä, yksi ominaisuus kerrallaan (kuvio 2).

```
body {
  background-color: #FFFFFF;
  font-family: 'Arial';
}

a {
  color: #2e8c00;
  font-size: 12px;
  text-decoration: underline;
}
```

*KUVIO 2. Body-elementille määritetään taustaväriksi valkoinen ja fontiksi Arial. Linkkielementeille määritetään väriksi tummanvihreä, fonttikooksi 12 pikseliä ja koristeluksi alleviivaus.*

Sanalla *cascading* eli ”putous” tarkoitetaan CSS:n pääperiaatetta eli periytymistä. Tyylilykset tulevat kolmesta pääasiallisesta lähteestä: selaimen omista oletustyyleistä, käyttäjän itse määrittämistä tyyleistä (eli käyttäjän muokkaamista selaimen oletustyyleistä) sekä kehittäjän määrittämistä tyyleistä. Kehittäjä voi tehdä tyylimäärittelyt kolmeen eri paikkaan: ulkoiseen CSS-tiedostoon, HTML-dokumentin alkuun (jolloin määrittelyt koskevat vain kyseistä dokumenttia) tai HTML-koodin sisään. (Mozilla Developer Network 2014, viitattu 8.3.2015.) Alla olevassa kuviossa (kuvio 3) on esitelty muut tavat paitsi käyttäjän tyylimäärittelyjen muokkaamisen, koska tyypillisesti nämä muutokset tehdään erilliseen CSS-tiedostoon selaimen luomassa profiilikansiossa.

```
<!DOCTYPE html>
<html lang="fi-FI">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
  <title>Sivusto.fi</title>
  <link rel="stylesheet" media="all" href="http://www.sivusto.fi/css/tyylit.css"/>
</head>
```

-----

```
<!DOCTYPE html>
<html lang="fi-FI">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
  <title>Sivusto.fi</title>
  <style>
    body {
      background-color: #FFFFFF;
      font-size: 12px;
      color: #000000;
    }
  </style>
</head>
```

*KUVIO 3. CSS-tyylimäärittelyjen mahdolliset sijainnit. Yllä on viitattu erilliseen CSS-tiedostoon, alla määrittelyt on tehty suoraan HTML-dokumentin alkuun.*

## 2.2.2 Media queryt

Media queryt eli mediakyselyt ovat CSS3:n mukana tullut uusi ominaisuus. Kun CSS2 mahdollisti tyylimäärittelyjen kohdistamisen erikseen eri medioille (kuten tietokoneen näytöille tai tulostusta

varten), CSS3:n media queryillä on mahdollista saada selville erilaisia tietoja käyttäjän päätelaitteesta ja kohdistaa tyylimääritykset niiden mukaan – ilman nuuskimista. Hyödyllisimmät tiedot web-kehittäjille ovat laitteen leveys ja korkeus, selaimen leveys ja korkeus, näytön resoluutio sekä laitteen orientaatio, eli pidetäänkö vaikkapa tablettia pysty- (portrait) vai vaakasuorassa (landscape). (Dhandhanian 2015, viitattu 8.3.2015.)

Mediakyselyt voidaan sijoittaa samaan CSS-tiedostoon kuin muutkin tyylimääritykset, sillä ne merkitään omaan, aaltosuluilla eroteltuun lohkoonsa (kuvio 4).

```
@media screen and (min-width: 480px) and (max-width: 600px) {  
  body {  
    background-color: #717b8a;  
    font-size: 14px;  
  }  
}
```

*KUVIO 4. Esimerkki media queryistä. Tässä määritellyt tyylit ovat voimassa vain, kun sivu esitetään näytöllä ja selaimen leveys on vähintään 480 mutta enintään 600 pikseliä.*

### 2.2.3 Frameworkit

Frameworkit eli asettelukehykset ovat kokoelma tiedostoja (HTML, CSS, JavaScript), joiden tarkoituksena on helpottaa rutiininomaisia web-suunnittelutehtäviä tekemällä niistä uudelleenkäytettäviä ”moduuleja”. Lyhyesti sanottuna asettelukehysten tärkein tavoite on häivyttää tarve rakentaa jokainen sivusto ”nollasta” ja vapauttaa kehittäjät työstämään ainoastaan niitä osia, jotka ovat yksilöllisiä tietyille projekteille. Samantapaisia elementtejä ei siis ole tarpeen keksiä uudelleen, vaan ne voi hyödyntää lähes samanlaisina useissa projekteissa. (Croft 2007, viitattu 9.3.2015.)

Tyypilliset asettelukehyksen komponentit ovat:

- Valmis CSS-koodi, jolla luodaan ”gridi” eli kehikko, jota käytetään elementtien sijoittelussa
- Typografia (fontit)

- Ratkaisuja selainyhteensopivuuden varmistamiseen
- Yhtenäiset CSS-luokat, joita käytetään eritasoisten komponenttien tyyliin

Asettelukehykset voi jakaa kahteen pääluokkaan: yksinkertaisiin ja täydellisiin. Yksinkertaisissa asettelukehyksissä on toteutettu lähinnä valmis responsiivinen kehikkojärjestelmä, joka mukautuu automaattisesti näyttökoon mukaan. Yksinkertaisia kehyksiä ovat esimerkiksi The 1140 CSS Grid, Toast ja 960 Grid System. Täydelliset kehykset puolestaan sisältävät (lähes) kaiken valmiina; muun muassa typografian, lomakkeet, painikkeet, ikonit ja HTML-templaattit. Esimerkkejä täydellisistä kehyksistä ovat muun muassa Bootstrap, Skeleton ja Foundation. (Awwwards-team 2015, viitattu 9.3.2015.)

Parhaimmassa tapauksessa asettelukehyksen käyttöönotto on erittäin nopeaa: periaatteessa kehittäjän ei tarvitse tehdä muuta kuin luoda uusi HTML-dokumentti, sisällyttää siihen jokin asettelukehykset ja alkaa personoida sitä. Personointi on helpoimmillaan kehyksen tyylien korvaamista omilla määrittelyillä – kuten vaikkapa vaihtamalla fontti tai värimaailma – ja sitä kautta voidaan keskittyä saamaan sillä hetkellä työn alla olevasta sivustosta yksilöllinen ja mieleenpainuva. (Croft 2007, viitattu 9.3.2015.)

Bootstrap on yksi suosituimmista responsiivisista asettelukehyksistä. Sen kehittivät Twitterillä työskentelevät Mark Otto ja Jacob Thornton vuonna 2011, ja se hyödyntää HTML-, CSS- ja JavaScript -kieliä. Bootstrap-pakkaus sisältää muokattavia aloitusteemoja ja sen etuja ovat hyvät laajat resurssit, kuten muun muassa laaja dokumentointi ja apuartikkelit, kolmansien osapuolten liitännäiset sekä erityiset ”teemarakentajat”. Bootstrap tukee uusimpia selaimia ja Internet Exploreria versiosta 8 eteenpäin (IE8 vaatii Respond.js -aputiedoston). Bootstrap on saatavilla MIT-lisenssillä. (Gerchev 2014, viitattu 22.3.2015.)

Foundation on toinen suosittu asettelukehykset, ja sitä on kehittänyt ZURB-niminen tuotesuunnitteluyritys vuodesta 2011 saakka. ZURB:in asiakkaisiin lukeutuvat muun muassa Facebook, eBay ja NYSE (New York Stock Exchange) (Schenker 2014, viitattu 7.5.2015).

”Oikeaa” asetelukehystä ei ole olemassa, sillä jokainen sivusto on erilainen. Kehysten käyttämisessä yleensäkin on sekä hyviä että huonoja puolia, joita kehittäjä voi punnita mielessään tutkiessaan seuraavan sivustonsa rakennustapaa. Kehysten hyviä puolia ovat siisti ja selkeä koodi, ratkaisut yleisiin CSS-ongelmiin, selainyhteensopivuus ja yhteinen merkitsemistapa, joka auttaa varsinkin silloin, kun sivustoa on rakentamassa tiimi eikä yksi henkilö. Huonoja puolia ovat jyrkähkö oppimiskäyrä ja mahdollisesti suuri määrä valmista mutta tarpeetonta koodia, muista resursseista puhumattakaan. Lisäksi oman ammatillisen oppimisen kannalta kehykset eivät ehkä ole se paras vaihtoehto; joku muu on tehnyt valmiin koodin, jota vain muokataan. (Awwwards-team, viitattu 9.3.2015.)

#### **2.2.4 Mobile first**

Mobile first – vaikka se ei varsinainen tekniikka olekaan, vaan pikemminkin responsiivisuuden työkalu – on ajattelutapa, jolla tarkoitetaan kehittämisen aloittamista sivuston mobiilinäkymästä. Termin esitteli web-kehittäjä Luke Wroblewski, ja hänen määritelmänsä perustuu kolmeen kohtaan:

- Mobiilikäyttäjien kasvu on tilaisuus tavoittaa enemmän ihmisiä kuin koskaan ennen
- Mobiililaitteiden rajoitteet pakottavat kehittäjät keskittymään olennaiseen – miten esittää kaikki oleellinen tieto pienellä näytöllä?
- Mobiililaitteissa käytettävät tekniikat – kuten paikannus- ja kosketusominaisuudet, kamera ja Bluetooth-yhteydet (Wroblewski 2009b, viitattu 10.3.2015) – luovat mahdollisuuksia innovoida, eivätkä ne välttämättä ole käytettävissä pelkällä tietokoneella

Mobile first siis sallii kehittäjien käyttää laitteiden kaikkia ominaisuuksia mahdollisimman monipuolisen käyttökokemuksen luomiseksi, kun web-sivustoja aletaan ensin kehittää uusimmalle tekniikalle. (Wroblewski 2009a, viitattu 10.3.2015.)

Kun sivustojen kehittäminen aloitetaan työpöytänäköisestä, ilmenee kaksi ongelmaa. Ensinnäkin kaikki CSS-tiedoston työpöytäversion tyylimäärittäisiin tehtävät muutokset vaikuttavat periytyvästi alempana dokumentissa määritettyihin mobiilimuotoiluihin CSS:n putous-luonteesta johtuen, mikä johtaa ylimääräiseen, pikkutarkkaan koodaukseen, johon uppoaa tarpeettoman paljon aikaa.

Alempana CSS-tiedostossa tehtävät tyylittelyt voi toki syrjäyttää !important-määrittelyllä, mutta ajan mittaan CSS-koodi on täynnä poikkeuksen poikkeuksia ongelmakohtiin ja tämä monimutkaistaa työskentelyä tarpeettomasti. Toiseksi mobiililaitteet lataavat jokaisen sivun sen työpöytäversion tyylimääritysten mukaan, koska ne ovat yleensä ensimmäisenä CSS-tiedostossa; ensin siis ladataan työpöytäversion tyylit, ja vasta sen jälkeen mobiiliversion. Tällä tavoin ladataan ylimääräisiä resursseja mobiiliversioon, mikä puolestaan hidastaa sivun ilmestymistä näytölle; jos sivulla on esimerkiksi taustakuva, mutta sitä ei näytetä mobiilissa vaan ainoastaan työpöytä- tai tablettikoossa, taustakuva ladataan joka tapauksessa. Tyypillisesti CSS-tiedostossa tyylimäärittelyt latautuvat ylhäältä alaspäin työpöytänäkymän määritysten ollessa ensin, sitten tabletin ja vasta lopuksi mobiiliin (kuvio 5).

#### DESKTOP FIRST

```
/*
DESKTOP
*/

.element {
  background: url('/media/images/background-image.jpg');
}

/*
MOBILE
*/
.element {
  background: #FFFFFF;
}
```

#### MOBILE FIRST

```
/*
MOBILE
*/

.element {
  background: #FFFFFF;
}

/*
DESKTOP
*/
.element {
  background: url('/media/images/background-image.jpg');
}
```

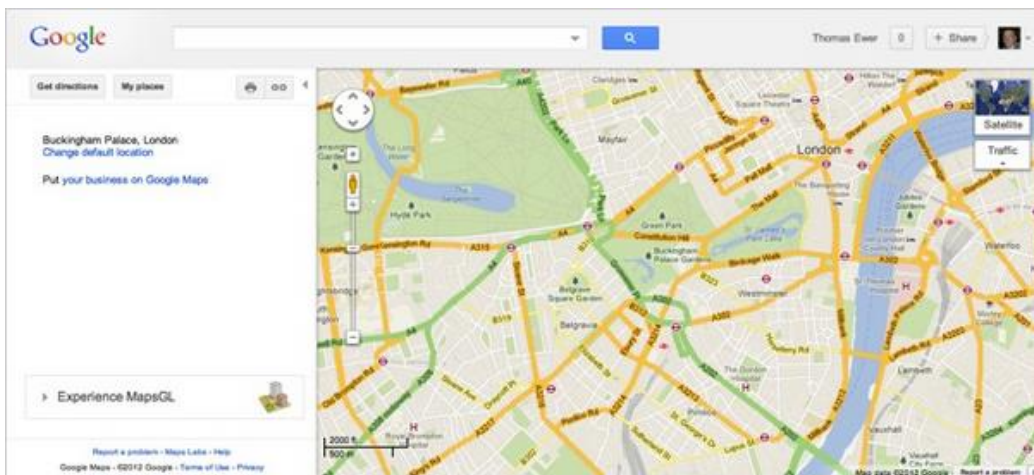
*KUVIO 5. Vasemmanpuoleisessa esimerkissä taustakuva on määritetty vain desktop-näkymälle, mutta koska se on määritetty ensin, se ladataan myös mobiilinäkymässä. Oikeanpuoleisessa esimerkissä taustakuva ladataan vain desktop-näkymään, koska se on määritetty mobiilimäärittelysten jälkeen.*

Mobile first -ajattelu vaikuttaa siis myös koodaustapaan; tyylimäärittelyt tehdäänkin järjestyksessä pienimmästä isompaan eikä toisin päin, ja ominaisuuksia lisätään sitä mukaa kun näytön koko kasvaa, jolloin ei ladata mitään tarpeetonta. (Baumann 2015, viitattu 11.3.2015.)

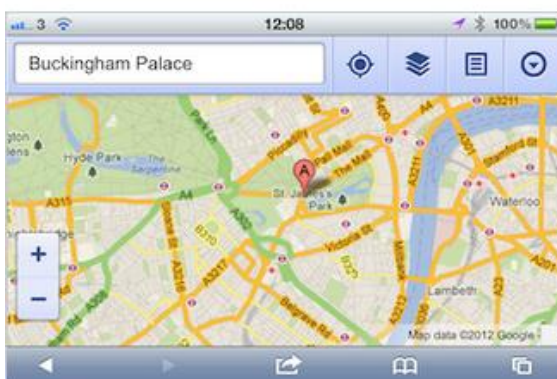


## 2.2.5 Responsiivisen web-suunnittelun kritiikkiä

Kaikki eivät kannata responsiivisten web-sivustojen kehittämistä. Tom Ewerin mielenkiintoisessa artikkelissa *5 Reasons Why Responsive Design Is Not Worth It* (Ewer 2012, viitattu 21.3.2015) kirjoittaja toteaa, ettei responsiivisuus kannata, ellei esimerkiksi web-aplikaation työpöytä näkymän sovittaminen mobiilikokoon ole käytännöllistä. Esimerkinä Ewer esittelee Google Maps -sivuston vuodelta 2012 (kuviot 6); työpöytä näkymässä kartta vei reippaasti yli puolet ruudun koosta ja vasemmalla oleva sivupalkki olisi luultavasti jäänyt näyttökoon pienentyessä kokonaan pois tai epäkäytännöllisen kokoiseksi samalle paikalle. Sen sijaan mobiilinäkymässä kartta olikin muutettu kokoleveäksi ja sivupalkki siirretty kokonaan näytön yläosaan.



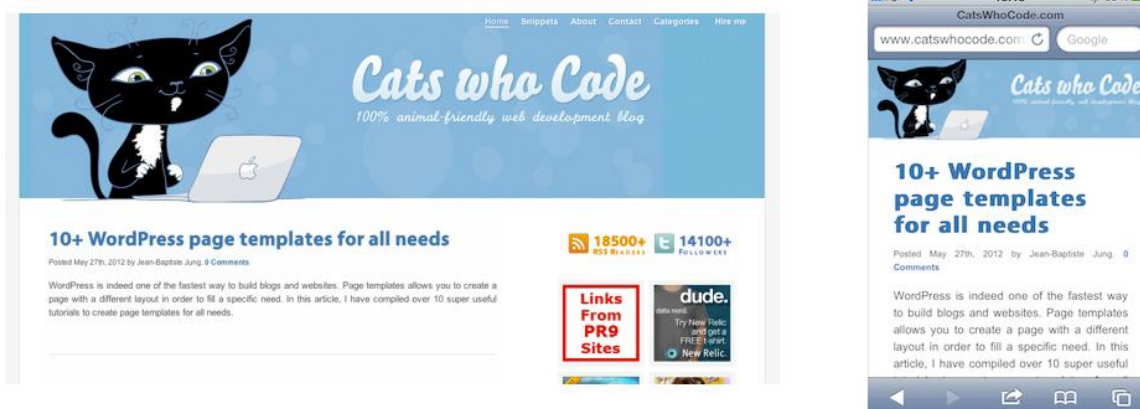
Desktop Display



Mobile display

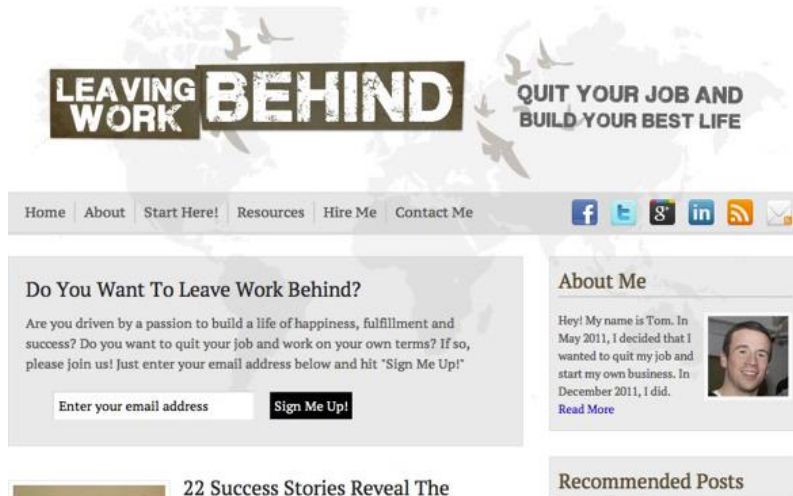
KUVIO 6. Google Maps -sivuston työpöytä- ja mobiilinäkymät vuodelta 2012 (Ewer 2012, viitattu 21.3.2015).

Ewerin ensimmäinen argumentti responsiivisuutta vastaan on se, että kehittäjien ensisijaisen tavoitteen tulisi olla käyttäjien odotuksiin vastaaminen, ja responsiivinen sivusto saattaa hämmentää käyttäjää. Ewer käyttää esimerkkinä yleistä blogisivustojen ”standardia”, jossa on tyypillisesti yläosa, sisältö yhdellä puolella ja sivupalsta toisella – siis malli, jota käytetään miljoonissa blogeissa, koska siinä navigointi on helppoa ja käyttäjät ovat tottuneet siihen. Tämän mallin responsiivinen toteutus todennäköisesti hävittää sivupalstan omalta paikaltaan, eikä se siis ole enää siellä, missä sen odottaisi olevan. Esimerkkinä tästä on artikkelissa käytetty *Cats who Code* -sivustoa, jossa työpöytä näkymän navigaatio ja sivupalsta ovat hävinneet mobiiliversiossa, mikä hämmentää käyttäjiä (kuvio 7). Sivupalsta oli itse asiassa siirretty varsinaisen sisällön alapuolelle, mutta navigaatio oli kadonnut kokonaan.



KUVIO 7. *Cats who Code* -sivuston työpöytä näkymä vasemmalla ja mobiilinäkymä oikealla (Ewer 2012, viitattu 21.3.2015).

Artikkelissa esitelly toinen argumentti on, että responsiivisuuden toteuttaminen maksaa enemmän kuin pelkkä sivuston staattinen ja mukautumaton työpöytäversio. Responsiivisen sivuston toteuttaminen vaatii luonnollisesti myös enemmän työtunteja ja saattaa tuottaa kehittäjille ylimääräisiä ongelmia. On helppo ajatella, että ei-responsiivinen sivusto on yhtä kuin kuollut sivusto, koska sitä ei voida käyttää mobiililaitteilla, mutta Ewerin kolmannen argumentin mukaan ei-responsiiviset sivustot ovat yleensä käyttökelpoisia myös mobiililaitteilla (kuvio 8).



KUVIO 8. Tom Ewerin oma ei-responsiivinen sivusto työpöytäkymässä (vasemmalla) ja iPhonella (oikealla). Sivusto on skaalattu iPhonen näytön kokoon. (Ewer 2012, viitattu 21.3.2015.)

Kuvion iPhone-näkymä näyttää oikeaa kokoaan monta kertaa pienemmältä, mutta Ewerin mukaan teksti on täysin luettavissa. Vaikkei se ei olisikaan, näkymä suurenee, jos iPhonen kääntää vaakatasoon. Ewerin sanoma onkin, että hyvin suunniteltu ei-responsiivinen sivusto on täysin käyttökelpoinen myös mobiilikokoisena, koska modernit mobiiliselaimet on kehitetty näyttämään työpöytäkokoisetkin sivut tehokkaasti.

Neljännessä argumentissaan Ewer toteaa, että vaikka responsiivisten sivustojen yksi päätarkoitus on keventää sivujen fyysistä kokoa mobiiliyhteyksiä silmällä pitäen, ei sivujen latautumisnopeudessa Ewerin mukaan yleensä ole eroja työpöytä- ja mobiilinäkömään välillä. Tämä johtuu lähinnä kehittäjien tavasta piilottaa elementtejä, mikä ei tarkoita etteivätkö ne latautuisi joka tapauksessa – tästä voidaan puhua huonona, mutta toisaalta yleisenä käytäntönä.

Viimeiseksi Ewer toteaa, että responsiivinen suunnittelu on kompromissien tekemistä, koska monesti suunnittelija pitää työpöytänäkömää sivun ”optimoituna” versiona ja mobiilinäkömää tämän karsittuna eli kompromettoituna versiona, jossa on jouduttu luopumaan tietyistä asioista esimerkiksi näytön koon vuoksi. Ewerin mukaan monet käyttäjät saattavat turhautua, koska heille esitetään erinäköinen sivusto mobiilissa, mikä saattaa jättää heille tunteen, ettei heillä ole kontrollia sivun suhteen.

Myös toisessa, hieman tuoreemmassa Stephan Jukicin blogiartikkelissa *How and Why Responsive Design Can Go Wrong* (Jukic 2013, viitattu 21.3.2015) todetaan, että joskus responsiivinen toteutus on tarpeetonta. Jukicin mukaan työpöytäkokoiseksi rakennettu sivusto toimii yleensä paitsi pöytä- ja kannettavilla tietokoneilla, mutta monesti myös jopa tabletilla niiden suurehkojen näyttöjen ansiosta. Tällöin jäljelle jäävät älypuhelinikäyttäjät ovat ainoa kohderyhmä, joka täytyy huomioida käytettävyyden kannalta, eikä silloinkaan ole tarpeen rakentaa koko sivustoa alusta loppuun uusiksi responsiivisena. Itse asiassa Jukic ehdottaa ratkaisuksi erillistä, mobiiliystävällistä versiota sivustosta mobiilikäyttäjää varten.

Myös Jucek mainitsee latausnopeuden responsiivisten sivustojen yhdeksi ongelmaksi ja painottaa, että kaikki tieto ei välttämättä ole oleellista sekä työpöytä- että mobiilikäyttäjälle, jolloin mobiilisivujen ongelmaksi saattaa muodostua periaatteessa kaiken mahdollisen mahdolluttaminen pieneen kokoon sen sijaan, että liika informaatio yksinkertaisesti vain karsittaisiin pois.

Lisäksi artikkelissa mainitaan asia, joka ei välttämättä käy monenkaan web-kehittäjän mielessä: jos sivustolla on maksettuja mainoksia, ovat mainostajat yleensä maksaneet siitä, että heidän ilmoituksensa näytetään sivuilla tietyssä koossa ja tietyssä paikassa. Sivujen mukautuessa mainoksia ei välttämättä voida näyttää halutussa kohdassa tai edes halutussa koossa, jolloin voidaan törmätä ongelmiin mainostajien kanssa.

Viimeisenä kohtana Jucek – kuten aiemmin myös Ewer omassa artikkelissaan – tuo esiin sen, että käyttäjä todella haluaa päättää itse. Ihmiset siis arvostavat valinnanvapautta ja tyypillisesti inhoavat sitä, että heidän kokemustaan rajoittavat pienet tekniset tekijät, joihin he eivät voi itse vaikuttaa. Ongelmana siis on, että koska responsiiviset sivustot mukautuvat laitteen koon mukaan, käyttäjä ei itse voi päättää, mikä version sivusta tämä näkee. Jotkut haluavat käyttää työpöytäkokoisista versiota myös tabletilla responsiivisen version sijaan.

### 3 OAMK:N TURVALLISUUSOHJEET -SIVUSTON TOTEUTUS

Tässä kappaleessa käyn läpi opinnäytetyöni käytännön toteutuksen eri vaiheet. Sivustolla ei ole vielä pysyvää fyysistä osoitetta, joten esittelen lopputuloksen paremmin kuvakaappauksilla kappaleessa 4.

Opinnäytetyöni aiheena on Oamk:n turvallisuusohjeiden kehittäminen responsiiviseksi web-sivustoksi. Oamk:n Liiketalouden yksikön turvallisuustiimi halusi nykyaikaistaa koulun turvallisuusohjeet verkkoon, koska niitä ei ollut aikaisemmin koottu yhteen paikkaan. Häätilanteessa tiedon nopea löytyminen on ensisijaista, joten responsiivinen web-sivusto on oiva ratkaisu Liiketalouden yksikössä, jossa valtaosalla opiskelijoista ja henkilökunnasta on käytössään jokin älylaite ja langaton internet.

Opinnäytetyöprosessi alkoi muutamalla palaverilla, joissa kävimme turvallisuusvastaavan sijaisen Pekka Ojalan kanssa turvallisuusohjeet läpi ja mietimme muutaman use casen eli käyttötapauksen, joiden kohdalla nopea ja helposti saatava apu olisi tarpeen. Use caset ovat tajuton henkilö, palava sähkölaite ja kaatunut tai liukastunut henkilö. Opinnäytetyön tavoitteena ei ollut rakentaa kokonaista sivustoa vaan pikemminkin ”demo”, jota voidaan mahdollisesti jatkokehittää ja ottaa käyttöön tulevaisuudessa.

#### 3.1 Suunnittelu

Koska kyseessä on ensisijaisesti mobiililaitteille tarkoitettu sivusto, tärkeimmät suunnittelun osa-alueet ovat käytettävyys ja tiedon tehokas esittäminen. Sivujen tulisi latautua nopeasti internetyhteystyypistä riippumatta, mikä tarkoittaa käytännössä sitä, ettei kuvia ole mielekästä käyttää ellei se ole ehdottoman välttämätöntä. Tästä syystä päädyin käyttämään ainoana kuvana Oamk:n logoa, joka ilmaisee sivuston ”omistajan” ja toimii samalla linkkinä etusivulle, kun sitä klikkaa tai koskettaa. Pienet näyttökoot pistivät miettimään tiedon mahdollisen tehokasta esittämistä, joten päädyin jQueryn UI-käyttöliittymäkirjaston Accordion-lisäosaan, joka mahdollistaa sisällön näyttämisen ja piilottamisen otsikoittain (kuvio 9). Accordion-elementin sisällön esittämistä hallitaan klikkaamalla tai koskettamalla. (jQuery UI 2015, viitattu 22.3.2015.)



KUVIO 9. jQuery UI:n Accordion -lisäosa. Haluttu sisältö jaetaan otsikoihin ja esitetään valittu otsikko kerrallaan. Esimerkissä "Onko rakennuksessa tai sen ulkopuolella välitöntä vaaraa?" on valittuna ja vain sen sisältö on näkyvillä, toinen otsikko on "suljettu". Otsikon edessä oleva nuoli on indikaattori laajennettavasta sisällöstä.

Päätin myös kokeilla mobile first -lähestymistapaa, koska en ole käyttänyt sitä aikaisemmin. Opinnäytetyön myötä tutustuin aiheeseen paremmin, ja halusin ehdottomasti kokeilla sitä käytännössä. Käytännössä tämä tarkoittaa sitä, että aloitin sivuston suunnittelun ja ohjelmoinnin pienimmästä näkymästä, eli kun leveys oli 480 pikseliä tai alle (kuvio 10). Halusin esittää sisällön mahdollisimman tehokkaasti, ja se onnistuikin hyvin kun käytettävissä oleva alue oli erittäin rajoitettu. Kun mobiilinäkymä oli suunniteltu, siirryin asteittain suurempaan näkymään ja sovitin sisällön niin, että se mukautuu luonnollisesti käytettävissä olevaan tilaan.



KUVIO 10. Sivuston mobiiliversion rautalankamalli.

### 3.2 Sivuston vaatimukset ja rakenne

Sivuston ensisijaiset vaatimukset ovat responsiivisuus, helppokäyttöisyys ja nopea latautuminen, varsinkin kun kyseessä on pääasiallisesti mobiilikäyttöön tarkoitettu sivusto. Graafisia tai ulkoasullisia vaatimuksia ei Oamk:n oranssin teemaväriin ja logon käyttämisen lisäksi annettu. Toimeksiantajan kanssa sovittiin, että testaus rajoitetaan muutamaa yleiseen, mutta erityyppiseen laitteeseen ja selaimiin.

Loppukäyttäjille näkyvä sivuston rakenne on yksikertainen: koulun logo (joka toimii myös linkkinä etusivulle), sivuston nimi ja navigointi ovat aina ensimmäisenä näkyvillä jokaisella sivulla (kuviot 10 ja 11). Kaikki sivut ovat päätason sivuja, jolloin navigointi pysyy yksinkertaisena ja selkeänä ilman alavalikkoja. Sivut on myös otsikoitu mahdollisimman yksiselitteisesti.



KUVIO 11. Sivuston desktop-version rautalankamalli.

Tekninen rakenne on myös pyritty tekemään selkeäksi: tiedostot on jaoteltu kansioihin (esimerkiksi "css" ja "javascript"), ja usein käytetyt elementit on sijoitettu omiin PHP-tiedostoihinsa, joita kutsutaan PHP:n include-toiminnolla. Tämä tekee koodista selkeämpää ja sivustosta helpommin ylläpidettävän, kun samoja muutoksia ei tarvitse tehdä useaan paikkaan.

### 3.3 Ulkoasu ja käytettävyys

Ulkoasu jatkaa sivuston yksinkertaista linjaa: oikealla palstalla on navigaatio, vasemmalla palstalla on logo ja sivuston nimi, sitten varsinainen sisältö ja alimpana "takaisin ylös" -painike, joka on esillä vain, jos sivua joutuu vierittämään.

Sivuston erikoisuutena ovat puhelinnumerolinkit. Kaikki sivustolta löytyvät puhelinnumerot ovat siis tel-muotoisia linkkejä, jotka toimivat eri tavalla käytettävästä laitteesta riippuen. Esimerkiksi



älypuhelimessa ”soita hätänumeroon” eli lähdekoodissa <a href="tel:112"> avaa puhelimen numerovalintaikkunan, jossa 112 on esivalittuna ja käyttäjän tarvitsee ainoastaan koskettaa ”soita”-kuvaketta. Muilla laitteilla, joissa ei ole puhelutoimintoa, sama linkki kysyy käyttäjältä ohjelmaa tai sovellusta, jolla linkki yritetään avata – siis samaan tapaan kuin sähköpostilinkkiä klikattaessa kysytään sähköpostiohjelmaa. Tietokoneilla ja tableteilla tel-muotoiset linkit voi avata esimerkiksi Skypellä.

Koska sivustolla on paljon numeroituja ohjeita, halusin tehdä niistä yksinkertaiset ja selkeät, mutta samalla myös tyylikkää. Numerot on korostettu oranssilla taustavärillä, ja ohjeiden numeroidut kohdat on erotettu toisistaan vaaleanharmaalla taustavärillä sekä pienellä marginaalilla (kuvio 12).

- 1 Soita hätänumeroon (112)
- 2 Asetu potilaan viereen
- 3 Paina käsivarret suorina potilaan rintakehää rintalastan alapuoliskon keskikohdalta. **Rintakehän tulee painua selvästi.**
- 4 Elvytysrytmin tulee olla **15:2**, eli 15 painallusta ja kaksi puhallusta, kun elvytät **yksin**. Kun elvyttäjiä on kaksi elvytysrytmi on **5:1**.
- 5 Puhalluselvytys aloitetaan **avaamalla hengitystiet**
- 6 Kallista potilaan päätä taaksepäin kohottamalla toisella kädellä leukaa ylöspäin ja paina toisella kädellä otsasta päätä taaksepäin.
- 7 Sulje otsaa painavan käden peukalolla ja etusormella potilaan sieraimet.
- 8 Aseta huulesi tiiviisti potilaan suuta vasten ja puhalla. Katso samalla että ilma menee perille seuraamalla potilaan rintakehää.
- 9 Keskeytä välillä elvytys ja tarkista potilaan pulssi

KUVIO 12. Elvytysohjeet tyylliteltynä.

### 3.4 Latausnopeuden optimointi

Mobiilisivustojen latausnopeus on oleellinen tekijä käyttökokemuksessa. Feedthebot.com -sivustolla olevassa artikkelissa *How to optimize CSS delivery* (Sexton 2015, viitattu 5.4.2015) todetaan, että ulkoisia CSS-tiedostoja tulisi olla vain yksi, eikä HTML-koodin sisäisiä CSS-määrittäjiä (inline) tulisi käyttää lainkaan. Myöskään @import-komentoa, jolla kutsutaan useita CSS-tiedostoja yhden tiedoston sisällä, ei saisi käyttää, sillä nämä kaikki hidastavat sivun latautumista ja lisäävät HTTP-kutsuja. Yksi kaikki määrittäjät sisältävä CSS-tiedosto on siis optimaalinen tapa tyyllitellä sivu, koska käytännössä se tarkoittaa ainoastaan yhden ”ylimääräisen” tiedoston lataamista. Lisäksi tämä tiedosto tallentuu selaimen välimuistiin, mikä puolestaan pienentää latautumisaikaa, jos käyttäjä vierailee sivustolla uudemman kerran.

Kuten aiemmin totesin, kokeilin opinnäytetyön toteutuksessa mobile first -lähestymistapaa. Kooditasolla tämä on toteutettu niin, että mobiiliversion määrittäjät ovat CSS-tiedostossa ensimmäisenä heti yleisten määrittäjä (värit, fontit, tehosteet jne.) jälkeen, jolloin kaikkien näkymien tyyllittelyt ladataan järjestyksessä pienimmästä (leveys enintään 480 pikseliä) suurimpaan (leveys vähintään 1200 pikseliä). Mobile first -metodi näkyisi luultavasti konkreettisemmin, jos sivustolla olisi käytetty enemmän kuvia, mutta sen ajatuksena on joka tapauksessa lähteä liikkeelle tiiviimmistä näkymästä ja edetä suurimpaan.

Latausnopeuden minimoimiseksi sivuston tyyli- ja JavaScript-tiedostot on tiivistetty yhteen CSS- ja yhteen JS-tiedostoon, ja kuvat on kompressoitu ilmaisella compressjpeg.com -sivustolla.

### 3.5 Navigointi

Navigointi on luonnollisesti tärkeä osa käyttäjäkokemusta. Etusivulla käyttäjältä kysytään ”mikä hätänä?” ja ohjataan yksiselitteisillä linkeillä tarvittavalle sivulle. Navigointi on työpöytänäkymässä sivun oikealla puolella ja mobiilinäkymässä sivun yläreunassa. Mobiilinavigoinnin toteutuksessa on käytetty Slicknav-nimistä jQueryn lisäosaa, joka on valmiiksi responsiivinen ja siinä on otettu kosketuspohjaiset laitteet erityisen hyvin huomioon.

### 3.6 Responsiivisuus

Koska kyseessä on hyvin pieni ja yksinkertainen sivusto, päätin jättää käyttämättä responsiivisia asettelukehyksiä. Sivustolla on käytetty kolmea breakpointia eli kohdelevyettä pikselinä, joiden kohdalla sivuston ulkoasu mukautuu:

- leveys vähintään 0 mutta enintään 480 pikseliä (ns. ”mobiilinäkymä”)
- leveys vähintään 768 pikseliä (ns. ”tablettinäkymä”)
- leveys vähintään 1200 pikseliä (ns. ”työpöytänäkymä”)

Responsiivisuus on toteutettu lähinnä määrittelemällä div-elementtien leveydet prosentteina, jolloin niiden leveys on aina tietty osuus selainikkunan koosta. Breakpointien kohdalla nämä leveydet hieman muuttuvat; esimerkiksi tabletti- ja mobiilinäkymässä (leveys 768 pikseliä ja alle) oikea palsta piilotetaan kokonaan, jolloin navigointi siirtyy sivun yläosaan. Lisäksi tekstikoot muuttuvat ja tekstirivien väli kasvaa käytettävissä olevan leveyden pienentyessä luettavuuden helpottamiseksi.

### 3.7 Testaus ja koodin validointi

Sivuston testaus on suoritettu seuraavilla laitteilla ja selaimilla (suluissa laitteen tyyppi ja käyttöjärjestelmä, selaimista on käytetty niiden uusinta saatavilla olevaa versiota):

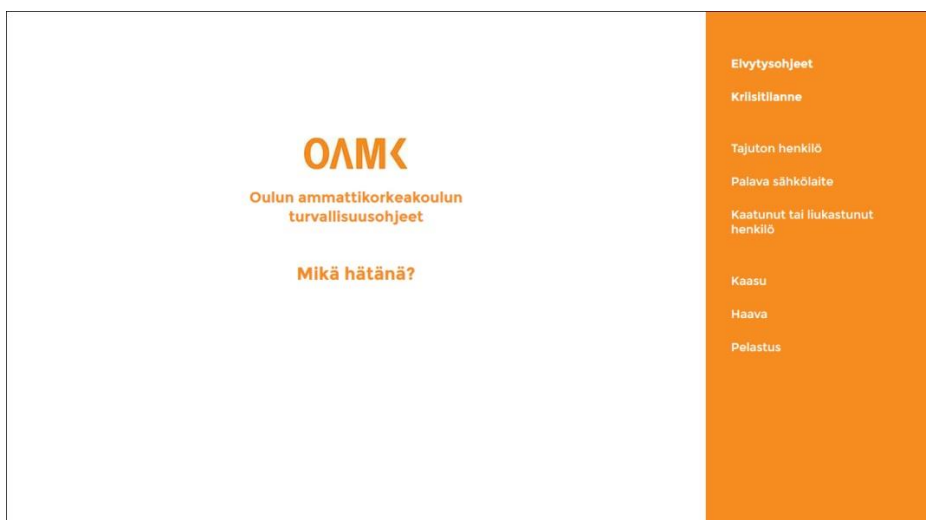
- Sony Xperia Z3 (älypuhelin, Android 5.0.2) – Firefox, Chrome
- Samsung Galaxy S3 (älypuhelin, Android 4.4.4) – Firefox, Chrome
- Samsung Galaxy Tab 4 (tabletti, Android 4.4.2) – Chrome
- Lenovo ThinkCentre M91p (pöytäkone, Windows 7) – Firefox, Chrome, Internet Explorer
- Lenovo IdeaPad Y50-70 (kannettava, Windows 8.1) – Firefox, Chrome, Internet Explorer
- MacBook Pro (kannettava, OS X 10.6.8) – Firefox, Chrome, Safari
- Apple iPad 2 (tabletti, iOS 8.1) – Safari
- Apple iPhone 4S (älypuhelin, iOS 7.1.2) – Safari
- Nokia Lumia 735 (älypuhelin, Windows Phone 8.1) – Internet Explorer

Sivusto on todettu täysin toimivaksi ja virheettömäksi kaikilla edellä mainituilla kokoonpanoilla.

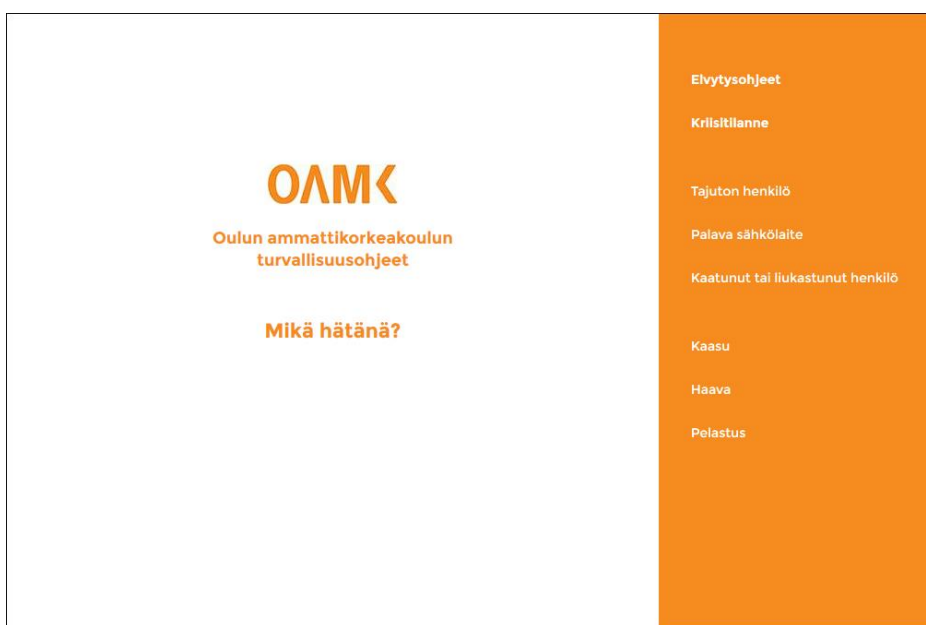
HTML-koodin validointi on toteutettu The W3C Markup Validation Servicellä, CSS puolestaan The W3C:n CSS Validation Servicellä. Ainoastaan CSS-validaattori antoi virheitä, mutta virheet johtuvat siitä, ettei itse validaattori ole täysin ajan tasalla.

## 4 JOHTOPÄÄTÖKSET

Tässä kappaleessa esittelen Oamk:n turvallisuusohjeet -sivuston, eli opinnäytetyöni käytännön osuuden. Sivustolla ei toistaiseksi ole fyysistä osoitetta, joten esittelen tuotokseni kuvakaappauksia käyttäen. Kaikkia sivuja en käy läpi, vaan ainoastaan ne, joissa responsiivisuus näkyy parhaiten.



KUVIO 13. Sivuston etusivu koossa 1280x800 pikseliä (työpöytänäkö).)



KUVIO 14. Sivuston etusivu koossa 1024x768 pikseliä (tablettinäkö).)



KUVIO 15. Sivuston etusivu koossa 320x480 pikseliä, navigaatio kiinni (mobiilinäkymä).



KUVIO 16. Sivuston etusivu koossa 320x480 pikseliä, navigaatio auki (mobiilinäkymä).

**OAMK** Oulun ammattikorkeakoulun turvallisuusohjeet

### Elvytysohjeet

- 1 Soita hätänumeroon (112)
- 2 Asetu potilaan viereen
- 3 Paina käsivarret suorina potilaan rintakehää rintalastan alapuoliskon keskikohdalta. **Rintakehän tulee painua selvästi.**
- 4 Elvytysrytmin tulee olla 15:2, eli 15 painallusta ja kaksi puhallusta, kun elvytät yksin. Kun elvyttäjiä on kaksi elvytysrytmi on 5:1.
- 5 Puhalluselvytys aloitetaan **avaamalla hengitystiet**
- 6 Kallista potilaan päätä taaksepäin kohottamalla toisella kädellä leukaa ylöspäin ja paina toisella kädellä otsasta päätä taaksepäin.
- 7 Sulje otsaa painavan käden peukalolla ja etusormella potilaan sieraimet.
- 8 Aseta huulesi tiiviisti potilaan suuta vasten ja puhalla. Katso samalla että ilma menee perille seuraamalla potilaan rintakehää.
- 9 Keskeytä välillä elvytys ja tarkista potilaan pulssi

Elvytystä jatketaan kunnes potilas virkoo tai paikalle saapuu ammattiapua.

- Elvytysohjeet
- Kriisitilanne
- Tajuton henkilö
- Palava sähkölaite
- Kaatonut tai liukastunut henkilö
- Kaasu
- Haava
- Pelastus

KUVIO 17. Elvytysohjeet koossa 1280x800 pikseliä (työpöytänäkö).

**OAMK** Oulun ammattikorkeakoulun turvallisuusohjeet

### Elvytysohjeet

- 1 Soita hätänumeroon (112)
- 2 Asetu potilaan viereen
- 3 Paina käsivarret suorina potilaan rintakehää rintalastan alapuoliskon keskikohdalta. **Rintakehän tulee painua selvästi.**
- 4 Elvytysrytmin tulee olla 15:2, eli 15 painallusta ja kaksi puhallusta, kun elvytät yksin. Kun elvyttäjiä on kaksi elvytysrytmi on 5:1.
- 5 Puhalluselvytys aloitetaan **avaamalla hengitystiet**
- 6 Kallista potilaan päätä taaksepäin kohottamalla toisella kädellä leukaa ylöspäin ja paina toisella kädellä otsasta päätä taaksepäin.
- 7 Sulje otsaa painavan käden peukalolla ja etusormella potilaan sieraimet.
- 8 Aseta huulesi tiiviisti potilaan suuta vasten ja puhalla. Katso samalla että ilma menee perille seuraamalla potilaan rintakehää.
- 9 Keskeytä välillä elvytys ja tarkista potilaan pulssi

Elvytystä jatketaan kunnes potilas virkoo tai paikalle saapuu ammattiapua.

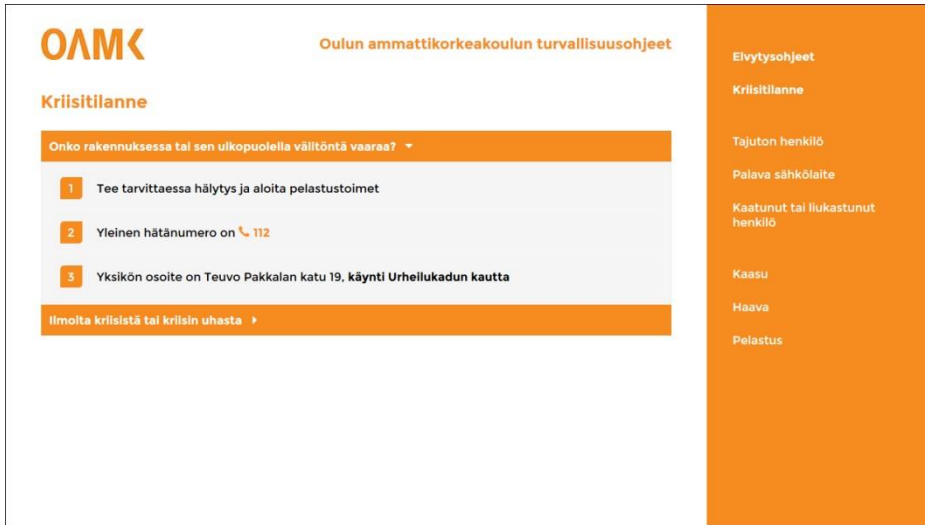
- Elvytysohjeet
- Kriisitilanne
- Tajuton henkilö
- Palava sähkölaite
- Kaatonut tai liukastunut henkilö
- Kaasu
- Haava
- Pelastus

KUVIO 18. Elvytysohjeet koossa 1024x768 pikseliä (tablettinäkö).

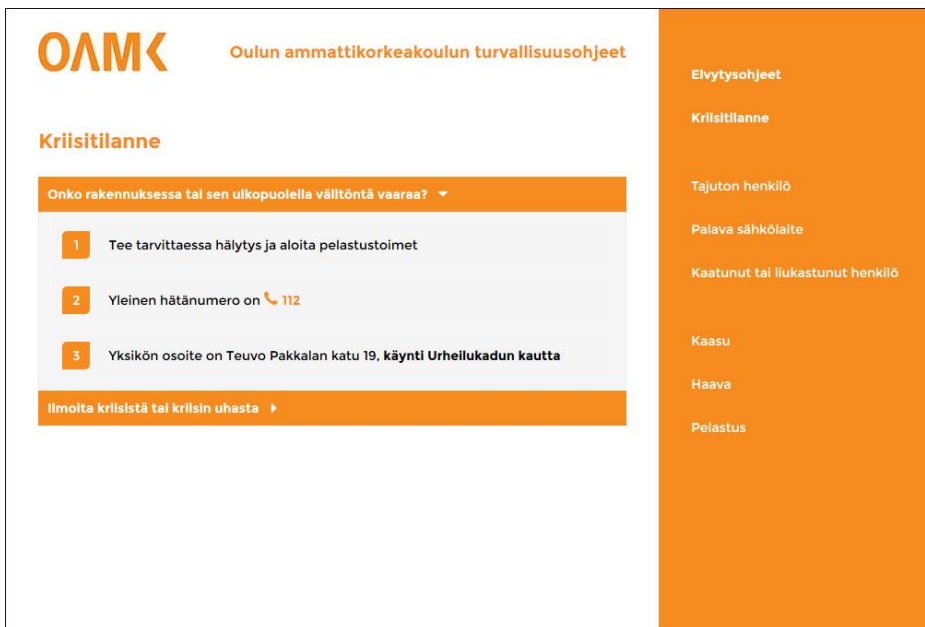


KUVIO 19. Elvytysohjeet koossa 320x480 pikseliä, navigaatio kiinni (mobiilinäkymä).





KUVIO 20. Kriisitilanne koossa 1280x800 pikseliä (työpöytäkymä).



KUVIO 21. Kriisitilanne koossa 1024x768 pikseliä (tablettinäkymä).



KUVIO 22. Kriisitilanne koossa 320x480 pikseliä, navigaatio kiinni (mobiilinäkymä).



*KUVIO 23. Takaisin ylös -painike mobiilinäkymässä. Painike ilmestyy, jos sivua vieritetään tarpeeksi paljon alas suhteessa laitteen tai näytön pikselikorkeuteen.*

## 5 POHDINTA

Kun aloitin opinnäytetyön tekemisen, olin jo rakentanut työssäni useita responsiivisia web-sivustoja. En kuitenkaan tiennyt responsiivisen web-suunnittelun historiasta paljoakaan, joten tämä oli hieno mahdollisuus tutustua sen alkuperään. Tutustuin myös teoriallasolla paremmin erilaisiin responsiivisiin asettelukehyksiin, mutta käytännön harjoittelu jäi harmikseni suunnitelmista poiketen toteutumatta, koska totesin opinnäytetyösivuston olevan liian suppea niiden käyttämiseen. Sen sijaan opin hirvittävän paljon mobile first -ajattelutavasta, ja innostuinkin kokeilemaan sitä käytännössä sivustoja ohjelmoidessani. En voi olla ajattelematta, miksen ole aikaisemmin aloittanut suunnittelua ja toteutusta mobiilinäkymästä – totta kai on järkevintä aloittaa suunnittelu sieltä, missä on eniten rajoituksia! Jatkossa aion ehdottomasti integroida mobile first -periaatteet omaan työhöni aina silloin kun mahdollista, varsinkin kun suurin osa yrityksemme asiakkaista haluaa responsiiviset web-sivustot. Ja kuten todettu, mobiilikäyttäjien määrä ja samalla heidän merkityksensä kuluttajina kasvaa jatkuvasti, eikä mobiiliystävällisyys ole vielä sillä tasolla jolle se voisi yltää.

Koin tarpeelliseksi etsiä myös näkökulmia responsiiviseen web-suunnitteluun kriittisesti suhtautuvilta, koska olin kiinnostunut mitä perusteita sen ”vastustamiselle” on keksitty. Henkilökohtaisesti olen sitä mieltä, etteivät nämä vastustavat perustelut ole kovin vakaalla pohjalla, mutta minusta oli joka tapauksessa tärkeää saada molemmat näkökannat mukaan. Itse lukeudun ehdottomasti responsiivisuuden kannattajiin, varsinkin kun olen käyttänyt sitä paljon työssäni ja huomannut, miten sivustoille voidaan suhteellisen pienellä vaivalla tuoda aivan uusi ulottuvuus. On totta, että responsiivisen sivuston suunnittelu ja rakentaminen vie enemmän sekä aikaa että rahaa, mutta kuten opinnäytetyössäni on aiemmin todettu, se on asia jota asiakkaat ovat oppineet odottamaan. Itse hämmästyin monesti, kun törmään internetissä jonkin yrityksen sivustoon, joka ei skaalaudu älypuhelimeeni; se antaa kyseisestä yrityksestä melko vanhanaikaisen kuvan. Toki ymmärrän sen, etteivät resurssit aina välttämättä riitä tai responsiivisuudelle ei nähdä perusteita, mutta ns. tavallinen kuluttaja saattaa olla omissa näkemyksissään mustavalkoisempi. Internet on kuitenkin suurilta osin markkinointikanava – ainakin nykyään – ja mainonnassa on aina pelattu mielikuvilla.

Varsinaisesta opinnäytetyösivuston tekemisestä en suoraan sanottuna oppinut kovinkaan paljon uutta, mutta kuten jo aiemmin totesin, mobile first -ajattelutapa teki minuun vaikutuksen. Sen kokeileminen ja siinä onnistuminen oli ehdottomasti koko opinnäytetyön parasta antia, ja olen oikeasti yllättynyt, miksei sille ole toistaiseksi annettu läheskään yhtä paljon arvoa kuin responsiivisuudelle, joka on oikeastaan vain tekniikka eikä suunnittelufilosofia. Kaikki kuitenkin lähtee suunnittelusta, ja ainakin minusta rajoittavimmasta tekijästä aloittaminen on nykyään kaikista loogisin tapa. Itse koen mobile first -filosofian melkein ainoaksi ongelmaksi sen, että asiakkaat odottavat yleensä ensimmäisenä nähtäväksi desktop-koon ulkoasumallia, eikä ns. mobiilinäkymälle ehkä osata antaa niin paljon painoarvoa. Toisaalta kokeneelle web-ohjelmoijalle ja -suunnittelijalle pelkän työpöytänäkömön ulkoasun soveltaminen pienempään kokoon käy monesti melko suoraviivaisesti, mutta eri asia on, kelpaavatko tehdyt ratkaisut asiakkaalle. Yllättävän monesti asiakas haluaa ehdottomasti melko epäolennaista sisältöä myös sivuston mobiiliversioon, eivätkä ammattilaisen perustelut välttämättä aina vakuuta niiden tarpeettomuudesta. Ongelma on siis se, että jos suunnittelu aloitetaan mobiiliversiosta edeten työpöytäversioon eikä toisinpäin, siihen kuluu asiakkaan mielestä liian paljon aikaa. Normi on tähän asti ollut nimenomaan eteneminen työpöytäversiosta mobiiliversioon, jolloin asiakas näkee ensin miltä sivusto näyttää ”täysikokoisena”, eikä välttämättä välitä niin paljon mitä sivustolle tapahtuu pienemmillä laitteilla. Onneksi poikkeuksiakin on, ja voin luonnollisesti puhua vain omista kokemuksistani. Lienee vain ajan kysymys milloin asiakkaatkin oivaltavat, ettei responsiivisuus tarkoita ainoastaan saman sisällön sovittamista pienemmälle ruudulle, vaan koko käyttökokemuksen mukauttamista. Mitä enemmän mobiilikäyttäjien määrä lisääntyy, sitä todennäköisempää on, että juuri se seuraava asiakas huomaa oman älylaitteensa kautta käyttökokemuksen tärkeyden – ehkä älylaitteilla on sittenkin muitakin rajoitteita kuin pelkkä näytön koko.

## 6 LÄHTEET

Awwwards-team. 2015. What are Frameworks? 22 Best Responsive CSS Frameworks for Web Design. Viitattu 9.3.2015, <http://www.awwwards.com/what-are-frameworks-22-best-responsive-css-frameworks-for-web-design.html>.

Baumann, D. 2015. Why you've got to start practicing mobile-first development. Viitattu 11.3.2013, <https://getflywheel.com/layout/start-practicing-mobile-first-development/>.

Croft, J. 2007. Frameworks for Designers. Viitattu 9.3.2015, <http://alistapart.com/article/frameworksfordesigners>.

Dhandhanian, A. 2015. Introduction to CSS Media Queries. Viitattu 8.3.2015, <http://www.htmlgoodies.com/beyond/css/introduction-to-css-media-queries.html>.

Ewer, T. 2012. 5 Reasons Why Responsive Design Is Not Worth It. Viitattu 21.3.2015, <https://managewp.com/5-reasons-why-responsive-design-is-not-worth-it>.

Gerchev, I. 2014. The 5 Most Popular Frontend Frameworks of 2014 Compared. Viitattu 22.3.2015, <http://www.sitepoint.com/5-most-popular-frontend-frameworks-compared/>.

Giarratano, A. 2015. Responsive Web Design Is Essential In 2015. Viitattu 3.5.2015, <http://www.business2community.com/digital-marketing/responsive-web-design-essential-2015-01205621>.

jQuery UI. 2015. Accordion. Viitattu 22.3.2015, <http://jqueryui.com/accordion/>.

Jukic, S. 2013. How and Why Responsive Design Can Go Wrong. Viitattu 21.3.2015, <http://www.sitepoint.com/responsive-design-can-go-wrong/>.

Kemp, S. 2015. Digital, Social & Mobile Worldwide in 2015. Viitattu 7.3.2015, <http://wearesocial.net/blog/2015/01/digital-social-mobile-worldwide-2015/>.

Kyrmin, J. 2015. What is CSS? Viitattu 7.3.2015, <http://webdesign.about.com/od/beginningcss/a/aa021607.htm>.

Leiniö, T. 2012. Mitä on responsiivinen design? Viitattu 7.3.2015, <http://www.sofokus.com/blogi/mita-on-responsiivinen-design/>.

Mozilla Developer Network. 2014. Cascading and inheritance - Web developer guide. Viitattu 8.3.2015, [https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Getting\\_started/Cascading\\_and\\_inheritance](https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Getting_started/Cascading_and_inheritance).

Schenker, M. 2014. Bootstrap vs. Foundation: Which Framework is Better? Viitattu 7.5.2015, <https://bootstrapbay.com/blog/bootstrap-vs-foundation/>.

Sexton, P. 2015. How to optimize CSS delivery. Viitattu 5.4.2015, <http://www.feedthebot.com/pagespeed/optimize-css-delivery.html>.

Six, J. M. 2013. Responsive Web Design and Accessibility. Viitattu 7.3.2015, <http://www.uxmatters.com/mt/archives/2013/04/responsive-web-design-and-accessibility.php>.

Soojian, C. 2015. A Brief History of Responsive Web Design. Viitattu 9.3.2015, <http://engage.syncoretech.com/marketing-technology-for-growth/bid/204297/A-Brief-History-of-Responsive-Web-Design>.

W3Schools. 2015a. CSS Introduction. Viitattu 8.3.2015, [http://www.w3schools.com/css/css\\_intro.asp](http://www.w3schools.com/css/css_intro.asp).

W3Schools. 2015b. CSS Syntax. Viitattu 7.3.2015, [http://www.w3schools.com/css/css\\_syntax.asp](http://www.w3schools.com/css/css_syntax.asp).

Web Education Community Group. 2013. Optimizing content for different browsers: the RIGHT way. Viitattu 7.3.2015, [https://www.w3.org/community/webed/wiki/Optimizing\\_content\\_for\\_different\\_browsers:\\_the\\_RIGHT\\_way](https://www.w3.org/community/webed/wiki/Optimizing_content_for_different_browsers:_the_RIGHT_way).

Wroblewski, L. 2009a. Mobile First. Viitattu 10.3.2015, <http://www.lukew.com/ff/entry.asp?933>.

Wroblewski, L. 2009b. Digital Product Platforms. Viitattu 10.3.2015, <http://www.lukew.com/ff/entry.asp?826>.