

# **BOOTSTRAP FRAMEWORK**

web-suunnittelun työkaluna

Veera Peltomäki

Opinnäytetyö  
Huhtikuu 2014  
Viestinnän koulutusohjelma  
Vuorovaikutteinen media

**TAMPEREEN AMMATTIKORKEAKOULU**  
**Tampere University of Applied Sciences**

## TIIVISTELMÄ

Tampereen ammattikorkeakoulu  
Viestinnän koulutusohjelma  
Vuorovaikutteinen media

VEERA PELTOMÄKI:  
Bootstrap framework  
web-suunnittelun työkaluna

Opinnäytetyö 66 sivua, joista liitteitä 0 sivua  
Huhtikuu 2014

---

Pienen näytön omaavat mobiililaitteet kasvattavat suosiotaan Internetin selaamisessa. Samaan aikaan modernit pelikonsolit ja SmartTV:t yleistyvät kuluttajien keskuudessa, jolloin Internetiä voi selata suuren näyttötarkkuuden omaavilla laitteilla. Responsiivinen web-suunnittelu vastaa nykypäivän vaatimukseen, jossa käyttäjät odottavat sivustoilta yhdenmukaista käyttökokemusta, päätelaitteesta riippumatta.

Tämä opinnäytetyö käsittelee responsiivisen Bootstrap frameworkin valintaa web-suunnittelun työkaluksi. Opinnäytetyössä tutkitaan mitä oikeastaan tarkoittaa responsiivinen web-suunnittelu ja sen keskeisiä tekniikoita. Sen jälkeen tutustutaan mikä on Bootstrap framework ja miten se vastaa responsiivisen suunnittelun vaatimukseen.

Opinnäytetyön lopuksi uudistetaan juhlapelit.net web-sivuston vastaamaan tämän päivän responsiivisia vaatimuksia. Juhlapelit.net on Wordpress-julkaisujärjestelmän päälle rakennettu web-sivusto, jonka tarkoituksena on tarjota ilmaisia peli-ideoita ja ajanvietettä aikuisten juhliin.

---

Avainsanat: bootstrap, framework, responsiivinen web-suunnittelu, mobile first, HTML, CSS, media queries

## **ABSTRACT**

Tampereen ammattikorkeakoulu  
Tampere University of Applied Sciences  
Media Programme  
Interactive Media

**VEERA PELTOMÄKI:**  
Bootstrap framework  
as a tool for web design

Bachelor's thesis 66 pages, appendices 0 pages  
April 2014

---

Mobile devices with small screens are becoming more popular for browsing the web. At the same time modern game consoles and SmartTV's are getting more common among consumers, which allows them to access web with big resolution devices. Responsive web design meets today's requirements where users are expecting consistent user experience, regardless of the terminal device.

This Bachelor's Thesis covers how Bootstrap framework works as a web design tool for responsive websites. The thesis investigate what responsive web design really means and what key techniques it covers. After that the thesis finds out what Bootstrap framework is and how does it respond for requirements of responsive web design.

End of the thesis will be rebuild juhlapelit.net web site to reply for today's responsive requirements. Juhlapelit.net is a Wordpress website, which purpose is to provide free game ideas and entertainment for a party for adults.

---

Key words:

bootstrap, framework, responsive web design, mobile first, HTML, CSS, media queries

## SISÄLLYS

1	JOHDANTO.....	7
2	RESPONSIIVINEN WEB-SUUNNITTELU .....	8
2.1	Responsiivisen web-suunnittelun lähtökohdat.....	8
2.1.1	Mobiili Internet .....	9
2.1.2	Näyttölaitteiden tarkkuus .....	10
2.2	Mobile first .....	12
2.2.1	Responsiivinen web-sivusto VS. mobiilisovellus.....	12
2.2.2	Mitä hyötyä mobile first tarjoaa? .....	13
2.3	Responsiiviset tekniikat .....	13
2.3.1	Mukautuva grid-järjestelmä .....	13
2.3.2	Mukautuvat kuvat ja media.....	15
2.3.3	Media queries .....	16
3	BOOTSTRAP - FRAMEWORK .....	19
3.1	Mitä framework tarkoittaa? .....	19
3.1.1	Suosittu Bootstrapin kilpailijat.....	19
3.2	Mikä on Bootstrap?.....	20
3.3	Käyttöluvat.....	20
3.4	Bootstrap eri julkaisujärjestelmissä .....	21
4	BOOTSTRAPIN KÄYTTÖ .....	22
4.1	Tiedostojen latausmuodot .....	22
4.1.1	Source code .....	22
4.1.2	Tuotantovalmis Bootstrap .....	22
4.1.3	Sass.....	23
4.1.4	Bootstrap CDN.....	23
4.2	Selain- ja laitetuki .....	24
4.3	Yleistä tekniikoista .....	25
4.3.1	Grid -järjestelmä .....	25
4.3.1.1	Kohdistetut palstat .....	28
4.3.1.2	Palstajärjestys .....	29
4.3.2	Typografia.....	29
4.3.3	Painikkeet.....	30
4.3.4	Kuvat.....	31
4.3.5	Lomakkeet.....	32
4.3.5.1	Peruslomake.....	33
4.3.5.2	Inline lomake .....	34
4.3.5.3	Horisontaalinen lomake .....	35

4.3.6	Taulukot .....	36
4.3.6.1	Perusmuotoilut .....	36
4.3.6.2	Hover-efekti .....	37
4.3.6.3	Responsiivinen taulukko .....	37
4.3.6.4	Rivin asiayhteys .....	38
4.3.7	Vaihtoehtoisen sisällön näyttäminen .....	38
4.4	Navigaatiot .....	39
4.4.1	Navigaation määriteluokat .....	39
4.4.2	Navigointipalkki .....	41
4.5	Komponentit .....	43
4.6	jQuery lisäosat .....	44
4.7	Kustomointi web-projektiin .....	44
4.7.1	Kustomoitujen tiedostojen lataaminen .....	44
4.7.2	Bootstrapin tyylien ylikirjoittaminen .....	45
5	BOOTSTRAP - SWOT ANALYYSI .....	47
5.1	Vahvuudet ja mahdollisuudet .....	47
5.2	Heikkoudet ja uhat .....	48
5.3	Yhteenvedo: Miksi projektiin kannattaa valita Bootstrap? .....	49
6	WEB-SUUNNITTELUN TRENDIT .....	51
6.1	Responsiiviset frameworkit .....	51
6.2	Flat design .....	52
6.3	Typografia .....	52
6.4	Single page -suunnittelu .....	53
6.5	Videot elävöittävät web-sivustoa .....	53
7	JUHLAPELIT.NET SIVUSTON UUDISTUS .....	54
7.1	Projektin lähtökohdat .....	54
7.1.1	Tekniikan valinta .....	55
7.2	Suunnittelu .....	57
7.3	Toteutus .....	58
7.3.1	Alustus .....	58
7.3.2	Templaattien rakennus .....	59
7.3.3	Ulkoasun kustomointi .....	59
7.4	Lopputulos .....	60
7.4.1	Analytiikka .....	61
8	POHDINTA .....	64
	LÄHTEET .....	65

**LYHENTEET**

WAP	Wireless Application Protocol
W3C	The World Wide Web Consortium
CSS	Cascading Style Sheet
IE	Internet Explorer selain
BS	Bootstrap
BS3	Bootstrap, versio 3
BS2	Bootstrap, versio 2
WP	Wordpress

## 1 JOHDANTO

Pienen näytön omaavat mobiililaitteet kasvattavat suosiotaan Internetin selaamisessa. Samaan aikaan modernit pelikonsolit ja SmartTV:t yleistyvät kuluttajien keskuudessa, jolloin Internetiä voi selata suuren näyttötarkkuuden omaavilla laitteilla. Responsiivinen web-suunnittelu tarkoittaa näytön tarkkuuteen mukautuvaan suunnittelua, jossa lopputuotteen kokoa ei voida tarkkaan määrätä (Marcotte 2012, 1). Tämän vuoksi responsiivinen suunnittelu vastaa hyvin nykypäivän vaatimuksiin, jossa käyttäjät odottavat web-sivustoilta yhdenmukaista käyttökokemusta, päätelaitteesta riippumatta (Datta A, 2012).

Tämä opinnäytetyö käsittelee responsiivisen Bootstrap frameworkin valintaa web-suunnittelun työkaluksi. Bootstrap tarjoaa pohjan web-sivustoille, jonka avulla suunnittelijoiden ei tarvitse rakentaa sivuja tyhjästä. Hyväksi havaittu koodi kierrätetään, jolloin vähennetään huomattavasti työtä ja säästetään aikaa (Awwwards, 2013).

Opinnäytetyössä tutkimme mitä oikeastaan tarkoittaa responsiivinen web-suunnittelu ja sen keskeisiä tekniikoita. Sen jälkeen tutustumme miten Bootstrap framework vastaa responsiivisen web-suunnittelun vaatimuksiin. Käymme läpi kuinka web-sivuston rakentamiseen käytettyä aikaa voidaan Bootstrapin grid-järjestelmän ja määriteluokkien avulla minimoida, jotta sivustosta saadaan tehtyä responsiivinen kustannustehokkaasti.

Lopuksi uudistamme juhlapelit.net web-sivuston vastaamaan tämän päivän responsiivisia vaatimuksia, Bootstrappia käyttäen. Juhlapelit.net on Wordpress-julkaisujärjestelmän päälle rakennettu web-sivusto, jonka tarkoituksena on tarjota ilmaisia peli-ideoita ja ajanvietettä aikuisten juhliin. Sivusto on julkaistu jo vuonna 2011, mutta se on aiemmin suunniteltu käytettäväksi vain työpöytäkoneella.

## 2 RESPONSIIVINEN WEB-SUUNNITTELU

Tässä luvussa käsitellään responsiivisen web-suunnittelun ja mobiililaitteiden Internetin historiaa, mobile first suunnittelua sekä näyttölaitteiden kehitystä. Sen jälkeen käydään läpi kolme avaintekniikkaa responsiivisuuden toteuttamiseen. Tekniikoita läpikäydessä keskitytään perusteisiin sekä yksinkertaisiin koodiesimerkkeihin.

### 2.1 Responsiivisen web-suunnittelun lähtökohdat

Responsiivinen web-suunnittelu tarkoittaa näytön tarkkuuteen mukautuvaan suunnitteluun, jossa lopputuotteen kokoa ei voida tarkkaan määrätä (Marcotte 2012, 1). Marcotte (2012, 3) käyttää esimerkkinään taiteilijaa, joka valitsee ensimmäiseksi kanvaasin, eli paperin tai kankaan, johon työnsä maalaa. Jokainen siveltimen veto on tarkkaan mietitty suhteessa taitelijan kanvaasin kokoon. Toisin kuin taitelijalla, web-suunnittelijan kanvaasi on selainikkuna. Selainikkunan koko vaikuttaa käytössä olevaan pinta-alaan ja sen kautta sisällön ja sisältöelementtien luettavuuteen ja näkyvyyteen (Marcotte 2012, 3). Tämän vuoksi responsiivinen web-suunnittelu on oikeaoppinen lähestymistapa suunniteltaessa web-sivustoa.

Terminä responsiivinen web-suunnittelu on melko uusi ilmiö. Marcotte kirjoitti ensimmäisenä aiheesta vuonna 2010 Responsive Web Design -artikkelissaan, List Apart -sivustolla. (Frain 2012, 10.) Termi yleistyi käytössä hänen samannimisen kirjan myötä seuraavan vuoden kuluessa. Marcotte (2010) määritteli artikkelissaan kolme olemassa olevaa tekniikkaa yhden yhtenäisen lähestymistavan alle, ja nimesi tämän responsiiviseksi suunnitteluksi. Nämä kolme tekniikkaa ovat mukautuva grid-järjestelmä, mukautuvat kuvat ja media sekä media queries.

Marcotte ei vain esitellyt uutta tapaa toistaa sivustoja pienillä näyttölaitteilla, vaan hän loi kokonaan uuden lähestymistavan nykyiselle web-suunnittelulle. Hänen mukaansa web-sivustoa tulisi suunnitella aluksi pienemmille näyttötarkkuuksille, ja vasta sen jälkeen ulkoasu ja sisältö mietitään suuremmassa mittakaavassa. (Frain 2012, 11.) Tätä lähestymistapaa kutsutaan nykyään mobile first suunnitteluksi.



Frain (2012, 33) haluaa korostaa, ettei responsiivinen suunnittelu ole mikään maaginen itsestäänselvyys, vaan sen käyttöönotto tulisi päättää sivuston lähtökohtien pohjalta. Lähtökohdissa punnitaan mm. sivuston budjetti, kohderyhmä ja tarkoitus. Toisaalta, vaikka sivuston lähtökohdat eivät suoranaisesti tukisi responsiivista suunnittelua, Frainin (2012, 33) omien kokemusten mukaan sen toteuttaminen tarjoaa lähes aina paremman käyttökokemuksen, kuin kiinteälevyinen sivusto.

### **2.1.1 Mobiili Internet**

Virallisesti mobiililaitteiksi ei lasketa ainoastaan puhelimia, vaan kaikki kannettavat laitteet. Näitä laitteita ovat esimerkiksi tabletit, kannettavat tietokoneet ja pelikonsolit, lukulaitteet, kamerat. (Viswanathan.) Tässä opinnäytetyössä mobiililaitteella tarkoitetaan kuitenkin älypuhelimia, sillä termi on yleistynyt tarkoittamaan kyseisiä laitteita alan sanastossa (johdannainen englanninkielisestä sanasta ”mobile phone”).

1990-luvun lopulla kehitetty WAP (Wireless Application Protocol) -teknologia oli ensimmäinen tapa selata Internetiä mobiilisti. Muutama vuosi sisällä sen kehittämisen jälkeen mm. Nokia ja Blackberry alkoivat tiuhaan julkaista markkinoille WAP -tuellisia laitteita. Näiden laitteiden suurin ongelma oli niiden kömpelöt selaimet, joissa oli puutteita mm. CSS- ja JavaScript-tuen, standardien sekä siirtonopeuden kanssa. Selaimien puutteiden vuoksi ensimmäiset mobiilisivustot olivat yksinkertaisia, rankalla kädellä karsittuja HTML-sivuja. (Datta 2012.)

Applen julkaisi ensimmäisen iPhoneen kesällä 2007, joka tarjosi oikeasti käyttäjäystävällisen tavan selata Internetiä mobiililaitteella. Pian Applen esimerkkiä seurasi moni muu puhelinvalmistaja. (Frain 2012, 22.) Uusien älypuhelin- ja tablettimallien tullessa markkinoille web-kehittäjät alkoivat rakentaa erillisiä mobiilisivustoja, jotka eivät käyttäneet samoja tiedostoja perinteisen, tietokoneella käytettävän pääsivuston kanssa. Kahden eri sivuston ylläpito oli työlästä, eikä usein mobiilisivuston tekniikka ollut yhtä kehittynyt kuin pääsivustolla. (Datta 2012.)

Responsiivinen web-suunnittelu mukauttaa pääsivuston sisällön mobiilikäyttäjän näytön tarkkuuteen. Tämä poistaa tarpeen kahden eri sivuston ylläpitoon. Kun responsiivisen

web-sivuston suunnittelu aloitetaan mobile first -konseptilla, voidaan mobiilikäyttäjälle tarjota kokonaisvaltainen käyttökokemus.

### 2.1.2 Näyttölaitteiden tarkkuus

Pienen näytön omaavat mobiililaitteet kasvattavat suosiotaan Internetin selaamisessa. Samaan aikaan modernit pelikonsolit ja SmartTV:t yleistyvät kuluttajien keskuudessa, jolloin Internetiä voi selata suuren näyttötarkkuuden omaavilla laitteilla. Tämän vuoksi web-sivujen suunnittelussa on alettu huomioimaan voimakkaammin sivujen toimivuus paljon vaihtelevissa näyttöjen tarkkuuksissa. Vertaillen W3Counterin (2014) tilastoja yleisimmin käytettyjen näyttötarkkuuksien osalta, viiden vuoden aikavälillä, tämän ilmiön voi osoittaa todeksi.

TAULUKKO 1. Näyttölaitteiden tarkkuus tammikuussa 2009

Sija	Tarkkuus (px)	Osuus (%)
1	1024x768	40.15
2	1280x800	16.54
3	1280x1024	15.10
4	1440x900	6.49
5	800x600	5.18
6	1680x1050	4.61
7	1152x864	3.24
8	1920x1200	1.45
9	1280x768	1.43
10	1280x960	1.09
	MUUT	4.72

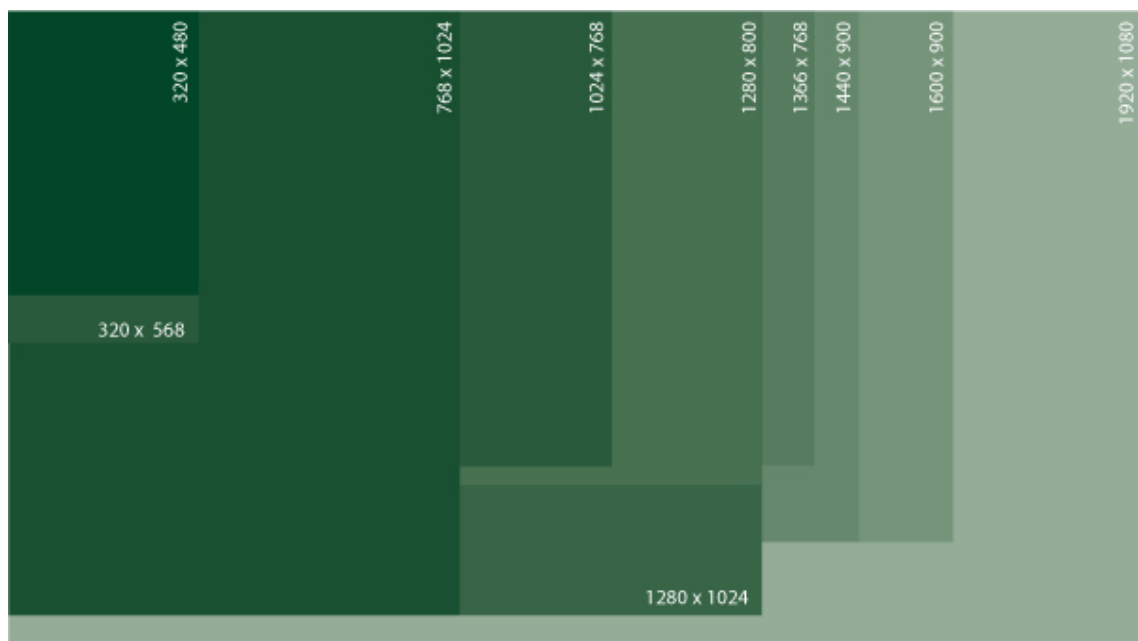
TAULUKKO 2. Näyttölaitteiden tarkkuus tammikuussa 2014

Sija	Tarkkuus (px)	Osuus (%)
1	1366x768	20.74
2	1024x768	9.61
3	1280x800	9.35

4	1920x1080	6.63
5	1440x900	5.44
6	768x1024	5.26
7	1280x1024	5.22
8	1600x900	4.14
9	320x480	2.90
10	320x568	2.81
	MUUT	27.9

Vuonna 2009 Internetin selaamiseen eniten käytetty näyttötarkkuus oli 1024x768 pikseliä, sen osuus markkinoista oli jopa 40.15 prosenttia. Viisi vuotta myöhemmin kyseinen näyttötarkkuus oli tippunut vain toiselle sijalle, mutta samalla menettänyt 3/4 markkinaosuudestaan. Osa tästä markkinaosuudesta on siirtynyt mobiililaitteille ja kasvaville näyttötarkkuuksille. Käytetyimmät pienet näyttötarkkuudet, joihin myös iPhoneen näytön tarkkuus lukeutuu, löytyvät sijoilta 9. ja 10.

Toinen asia jonka tilastoista voimme todeta, on näyttötarkkuuksien selkeä monipuolistuminen (MUUT markkinaosuusosuus). Kun vuonna 2009 muiden osuus oli vain vaivaiset 4.72 prosenttia, viiden vuoden kuluessa tämä luku on lähes kuusinkertaistunut.



KUVA 1. Top 10 näyttötarkkuudet 2014

## 2.2 Mobile first

Ennen mobiili Internetin kehitystä, web-sivustot suunniteltiin pääasiassa vain työpöytä-näyttöille. Vaikka mobiili Internet alkoi yleistyä, silti suunnittelijat mieltivät sivuston toimintaa mobiilissa vasta sen jälkeen, kun sivusto oli suunniteltu isolle näytölle, jos edes silloinkaan. Vain harvat käyttivät mobiililaitteitaan Internetin selaamiseen hitaan yhteyden ja kömpelön käyttökokemuksen vuoksi. Ne jotka käyttivät, saivat eteensä vain epäkäytännöllisiä web-sivustoja, joiden selaaminen oli hankalaa. (Wroblewski 2011, 1.) Kasvavien mobiilimarkkinoiden myötä suunnittelijoiden piti alkaa huomioida myös mobiilikäyttäjät; millainen sisältö ja käyttäjäkokemus heille web-sivustolla halutaan tarjota. Tämän pohjalta syntyi mobile first suunnittelu.

Mobile first suunnittelun tarkoittaa, että web-sivuston rakenne ja toiminnallisuus suunnitellaan ensin pienimmän ja heikkotehoisimman päätelaitteen mukaan. Sen tavoite on tarjota mobiilikäyttäjälle mahdollisimman hyvä käytettävyyys sekä vain tärkeimmät ja tarpeelliset sisällöt (Wroblewski 2011, 50).

### 2.2.1 Responsiivinen web-sivusto VS. mobiilisovellus

Responsiivinen web-suunnittelu ei poista perinteisen mobiilisovelluksen tarvetta, sillä web-sivustolla ei ole pääsyä järjestelmän resursseihin samalla tavalla kuin sovelluksella. Järjestelmän resursseista sovellus voi lukea esimerkiksi käyttäjän yhteystietoja, kalentteria, kuva-albumeita, jne. Mutta jos kyseessä on normaali web-sivusto, joka ei vaadi pääsyä laitteen järjestelmän resursseihin, ei ole syytä toteuttaa sivustoa erikseen mobiilisovellukseksi.

Mobiilisovelluksen koodikieli riippuu puhelinvalmistajasta, kun taas web-sivuston HTML:llä voi mikä tahansa mobiiliselain lukea. Esimerkiksi Apple iOS sovellukset ovat tehty Objective C-kielellä, Google Android luottaa Javaan, Microsoft Windows Phone 7 lukee C#:ia, jne. Lisäksi käyttäjän pitää erikseen ladata sovellus ja sen päivitykset, millaista tarvetta taas responsiivisessa sivustossa ei ole. (Wroblewski L 2011, 15.)

### **2.2.2 Mitä hyötyä mobile first tarjoaa?**

Wroblewski (2011, 19) käyttää esimerkkinään web-sivustoa, jonka kohderyhmän näytön tarkkuudeksi päätetään 1024 x 768 pikseliä. Sivusto suunnitellaan kyseiseen kokoon, jossa sen sisältö, graafinen ulkoasu ja käytettävyys ovat viimeistelty. Käyttäjä saapuukin sivustolle iPhonellaan, jonka näytön tarkkuus on vain 320 x 480 pikseliä. Tällä iPhonen käyttäjällä on siis 80 prosenttia pienempi näyttötarkkuus kuin mitä kohderyhmälle on päätetty. Miten tälle kävijälle voidaan tarjota kokonaisvaltainen käyttökokemus vain murto-osalla alkuperäisen web-sivuston koosta? Mobile first suunnittelu nimenomaan pyrkii ehkäisemään tällaista ongelmaa.

Kun sivuston suunnittelu aloitetaan ottamalla huomioon pienemmät näyttötarkkuudet, ollaan jo pitkällä. Pienille näyttötarkkuuksille suunnittelu pakottaa suunnittelijan miettimään, mitä web-sivuston tulee tarjota mobiilikäyttäjälle. Kun käyttäjä tulee sivustolle työpöytäkoneellaan, hän todennäköisesti etsii sivustolta erilaista tietoa kuin mobiilikäyttäjä. Esimerkiksi ravintolan web-sivustolla mobiilikäyttäjälle tulisi tarjota helppo pääsy ruokalistaan ja hänen tulee voida nopeasti tarkistaa ravintolan aukioloajat. Työpöytäkäyttäjälle voidaan tarjota visuaalisempi kuva ravintolasta esim. suurien kuvituskuvien avulla.

## **2.3 Responsiiviset tekniikat**

Marcotten (2012, 9) määrittelyn mukaan responsiivinen suunnittelu terminä kattaa kolme, jo pitkään olemassa ollutta tekniikkaa. Tässä osiossa käsitellään näiden tekniikoiden perusteita.

### **2.3.1 Mukautuva grid-järjestelmä**

Kun Internet yleistyi 1990-luvulla, oli normaalia että web-sivut koostettiin taulukkora-kenteen päälle. Taulukot olivat ainoa tehokas tapa sitoa tekstiä ja grafiikkaa yhteen. Kun sivustoista alettiin rakentaa kehittyneempiä ja laajempia, taulukkora-kenteet muuttuivat monimutkaisemmiksi, sisäkkäin koostetuiksi taulukoiksi. Monimutkaisuus toi mukanaan ongelmia sivuston päivitettävyyteen, sillä merkintäkieli (markup) ei ollut kovin

selkeää. (Ma.Cro Mind 2013.) Kun taulukoiden tyyllittely tuli mahdolliseksi, taulukon ja sarakkeiden sarakeleveydet laskettiin usein prosentteina. Esimerkiksi koko taulukko oli 100 prosenttia leveä, eli koko näytön levyinen. Vasen sarake oli 20 prosenttia ja pääsisällön sarake oli 80 prosenttia taulukon kokonaisleveydestä. Tällöin sivusto mukautui käyttäjän selainikkunaan, näytön tarkkuudesta huolimatta.

CSS standardin julkaisu vuonna 1996 loi uuden suunnittelutavan web-sivustoihin, jota kutsuttiin Table-less suunnitteluksi (Ma.Cro Mind 2013). Taulukkorakenteiden sijaan suunnittelijat alkoivat rakentaa sivustoja `div` elementtien päälle. `div` elementit helpottivat web-sivuston sisällön löytämistä HTML-koodin seasta ja näin paransivat sivujen päivitettävyyttä.

Mutta ei hyvää ilman pahaa. Kun `div`-pohjainen suunnittelu syrjäytti taulukkorakenteet, prosentit muuttuivat kiinteästi määriteltyihin pikselileveyksiin, jotta web-sivuista saatiin enemmän printtimedian kaltaisia (Frain 2012, 61). Koska web-sivusto mediana on hyvin erilainen kuin esimerkiksi A5 kokoinen esite, pikselintarkka leveydenmäärittely toi ongelmia vaihtelevien näytön tarkkuuksien kanssa. Tämän vuoksi responsiivisessa web-suunnittelussa on palattu ajassa takaisin 1990-luvulla käytettyyn lähestymistapaan, prosentuaaliseen leveyden määrittelyyn.

Vaikka web-sivulla leveydet olisivat jo rakennettu kiinteisiin pikselileveyksiin, Marcotte (2012, 31) esittelee tähän kaavan, jolla pikselit saa helposti laskettua prosenteiksi:

$$\textit{kohde} / \textit{yhteys} = \textit{osamäärä}$$

Tässä laskukaavassa *kohde* on yhtä kuin kiinteän `div`:n leveys, joka muutetaan prosenteiksi. *Yhteys* tarkoittaa ulkoista leveyttä, johon `div`:n leveyttä verrataan. Kun *osamäärä* kerrotaan sadalla, saadaan tulo eli `div`:n leveys prosentteina.

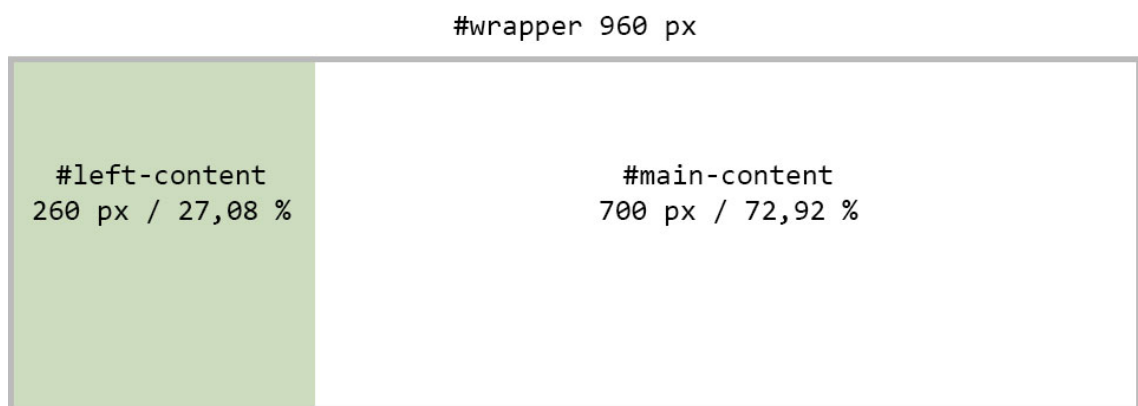
Alla on HTML ja CSS esimerkki yksinkertaisesta sivurakenteesta, josta on tarkoitus muuttaa `#left-content` ja `#main-content` leveydet prosenteiksi.

```
<div id="wrapper">
  <div id="left-content">...</div>
  <div id="main-content">...</div>
</div>
```

```
#wrapper{ width: 960px; }
#left-content{ width: 260px; }
#main-content{ width: 700px; }
```

Ensiksi lasketaan, kuinka monta prosenttia `#left-content` on sen isäntäelementin, `#wrapper` div:n leveydestä. Kaava on silloin seuraava:  $260 \text{ px} / 960 \text{ px} = 0,2708 * 100 = 27,08 \%$ . Koska muutettavia div elementtejä on vain kaksi, `#main-content` div:n prosentuaalisen leveyden voi laskea joko samalla kaavalla tai vähentämällä `#left-content` div:n tulon 100 prosentista. Molemmilla kaavoilla saadaan `#main-content` div:n leveydeksi  $72,92 \%$ .

```
#wrapper{ width: 960px; }
#left-content{ width: 27,08%; }
#main-content{ width: 72,92%; }
```



Kuva 2. Esimerkkisivun rakenne pikseleinä ja prosentteina

Itse suosin tapaa, jossa ulkoiselle div elementille, kuten `#wrapper`, määrätään kiinteä leveys pikseleinä. Jos se määritellään prosentteina, esim.  $100 \%$  näytön leveydestä, web-sivuston `#main-content` on isoilla näyttötarkkuuksilla häiritsevän leveä ja vaikealukuluinen, mikäli se sisältää tekstiä. Kuten Bootstrapissa, ulkoiselle div:lle kannattaa mieluummin määrittää eri näyttöleveyksiin omat breakpointsit (taitepisteet) media queriesin avulla, jossa sen leveys muutetaan näytön leveyteen sopivaksi.

### 2.3.2 Mukautuvat kuvat ja media

Kun sivustosta luodaan responsiivinen, pitää myös sen sisältämien kuvien osata suhteuttaa kokonsa näytön tarkkuuteen. Suosituin ja helpoin tapa on käyttää CSS:n `max-width`

ominaisuutta. Tällöin pitää muistaa, ettei `img` elementille saa määrittellä `width` ja `height` attribuutteja.

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

Samaa CSS ominaisuutta käyttäen myös muu sivustolta löytyvä media voidaan muuttaa näytön leveyteen mukautuvaksi.

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

Poikkeuksena mukautuvasta mediasta Frain (2012, 77) mainitsee `iframe` elementin, kuten sivustolle upotetut YouTube videot, joiden leveyden saa muutettua samalla `max-width` ominaisuudella, mutta korkeuden säätämiseen tarvitaan avuksi JavaScript lisäosaa. Tästä tekniikasta voi lukea tarkemmin Frainin kirjasta *Responsive Web Design with HTML5 and CSS* (2012, 26).

### 2.3.3 Media queries

Kun nykyaikaisten mobiililaitteiden kehitys kiihtyi, käyttäjät alkoivat odottaa sivustoilta yhdenmukaista käyttökokemusta, päätelaitteesta riippumatta (Datta 2012). Muun muassa tähän tarpeeseen kehitettiin tekniikka nimeltään ”media queries”.

Media queriesit ovat osa CSS3 -määrittelyä (W3C 2012) ja niiden tarkoitus on laajentaa `media` attribuutin toimintaa. Esimerkiksi `media="print"` kohdentaa tyylimäärittelyt tulostettavaa sivunäkymää varten. Media queriesit vievät tämän kohdentamisen uudelle tasolle, kun web-kehittäjille tarjottiin mahdollisuus kohdentaa tyylimäärittelyt lukuisiin laitekohtaisiin ominaisuuksiin, kuten näytön leveyteen tai kuvasuhteeseen.

Media queries määriteltiin Internet standardeja määrittelevän W3C konsortion suosittelemaksi standardiksi kesäkuussa 2012, mutta sen kehittäminen ja käyttö on aloitettu jo kauan ennen tätä suositusta. Ensimmäinen suositusehdotus (Candidate Recommendation) julkaistiin W3C:n web-sivustolla heinäkuussa 2002, eli liki kymmenen vuotta ennen virallisen standardisuosituksen päätöstä. Tästä seuraavat viisi vuotta oli hiljaista kehi-



tyksen osalta, kunnes vuonna 2007 iPhonen tulo markkinoille käynnisti kehityksen jatkumun viralliseen suositukseen saakka.

TAULUKKO 3. W3C:n Media queries päivityshistoria

Pvm	W3C julkaisu
19.6.2012	Media Queries Recommendation
26.4.2012	Proposed Recommendation
27.7.2010	Candidate Recommendation
15.9.2009	Candidate Recommendation
23.4.2009	Candidate Recommendation
15.10.2008	Last Call
6.6.2007	Candidate Recommendation
8.7.2002	Candidate Recommendation

Media queries on laajasti tuettuna kaikissa moderneissa selaimissa. Yleisesti vanhempiin selaimiin, kuten Internet Explorerin versioihin 7-8, media queriesin saa toimimaan JavaScript -pohjaisella lisäosalla. (Frain 2012, 35.)

TAULUKKO 4. Media queriesia tukevat selaimet

Selain	Versio
Firefox	3.6+
Safari	4+
Chrome	4+
Opera	9.5+
iOS Safari	3.2+
Opera Mobile	10+
Android	2.1+
Internet Explorer	9+

Yksinkertaisuudessaan media queriesilla määritellään tyylit halutuille laite- ja näyttökokonaisuuksille. Niiden avulla web-sivustolle voidaan ladata jonkin tietty tyylitiedosto HTML-dokumentin head:n sisällä. Alla olevassa esimerkissä sivustolle ladataan mobi-le.css niminen tyylitiedosto, mikäli näytön leveys on alle 768 pikseliä.

```
<link href="css/mobile.css" rel="stylesheet" media="screen and (max-width: 768px)" >
```

Vaihtoehtoisesti media queriesia voidaan käyttää suoraan sivuston päätyylitiedoston sisällä. Alla olevassa esimerkissä `.someDiv` piilotetaan, mikäli näytön leveys on alle 768 pikseliä.

```
@media screen and (max-width:768px){  
  .someDiv{ display:none; }  
}
```

`.somediv` voi sisältää esimerkiksi kuvituskuva, joka halutaan näyttää vain työpöytä- ja tablettikäyttäjälle. Koko `div:n` voi tällöin piilottaa mobiilinäkymästä, mikäli se ei ole sivuston sisällön kannalta oleellinen. Media queriesit tarjoavat siis suunnittelijalle apuvälineen, jolla hän voi hallita web-sivuston rakennetta ja sisältöä eri näyttötarkkuuksissa.

### 3 BOOTSTRAP - FRAMEWORK

Tässä osioissa tutustumme mitä Bootstrap ja framework ylipäätään tarkoittavat. Samalla käymme läpi BS:n käyttöehtoja, ja miten se taittuu erilaisiin julkaisujärjestelmiin.

#### 3.1 Mitä framework tarkoittaa?

Framework on käsitteellinen konsepti, joka tarjoaa ns. vakiopakkauksen web-sivuston rakentamiseen. Web-suunnittelijoiden tasolla framework tarkoittaa valmista HTML, CSS ja JavaScript tekniikoiden kirjastoa, jonka tarkoitus on tarjota helppo, standardeihin tähtäävä työkalu verkkototeutuksiin. (Awwwards, 2013.)

Useiden web-sivustojen rakenne on melko samankaltainen: navigaatio ja logo ovat tietyssä paikassa, käytössä on 12-palstainen grid-järjestelmä, jne. Frameworkit tarjoavat yhteisen pohjan web-sivustoille, jonka avulla suunnittelijoiden ei tarvitse rakentaa sivustoja aina tyhjästä. Hyväksi havaittu koodi kierrätetään, jolloin vähennetään huomattavasti työtä ja säästetään aikaa (Awwwards, 2013).

##### 3.1.1 Suositut Bootstrapin kilpailijat

Bootstrapin lisäksi suosittuja avoimen lähdekoodin responsiivisia frameworkkeja ovat mm. HTML5 Boilerplate, Foundation sekä Skeleton. Kaikki mainitut frameworkit pohjautuvat HTML5:seen.

Vuonna 2010 julkaistu HTML5 Boilerplate oli yksi ensimmäisistä avoimen lähdekoodin frameworkkeista, joka sisälsi responsiivisen, mobiiliystävällisen HTML5-templaatin (Gube 2013). Sen vahvuus kilpailijoihinsa nähden on kehittynyt tuki eri selaimille.

Foundation on suunniteltu web-sivustojen rakentamiseen mobile first lähestymistavalla (Gube 2013). Foundation mainostaa itseään maailman kehittyneimmäksi responsiiviseksi frameworkkeiksi, 12-palstaisen grid-järjestelmänsä vuoksi (Foundation, 2014). Bootstrapista löytyy paljon yhtymäkohtia Foundation grid-järjestelmään.

Skeleton on yksikertainen ja simppelempi responsiivinen framework web-sivustoille. Se sisältää vain responsiivisen, 12-palstaisen grid-järjestelmän, kevyet CSS-tyylien peruspohjat, media queriesit ja HTML5 tuen vanhemmille selaimille. (Gube 2013.) Skeletonin vahvuus kilpailijoihin nähden on sen yksikertaisuus; se ei sisällä CSS-tiedostojen lisäksi mitään ylimääräistä.

### 3.2 Mikä on Bootstrap?

Ensimmäinen versio Bootstrapista (aiemmin tunnettu nimellä Twitter Bootstrap) julkaistiin elokuussa 2011. Sen kehittäjinä toimi tuolloin vain kaksi miestä, Mark Otto ja Jacob Thornton (nimimerkeiltään @mdo ja @fat). Vaikka ensimmäinen versio BS:ta ei ollut vielä responsiivinen, Cochranin (2012, 2) mukaan siitä tuli suosittu huolellisen ja helposti muokattavan koodinsa ansiosta.

Tammikuussa 2012 julkaistu Bootstrap versio 2 otti suuren harppauksen eteenpäin, sillä se sisälsi huomattavasti enemmän laajennuksia sekä responsiivisen taiton. Responsiivisuus oli toteutettu mukautumaan kolmeen suurimpaan päätelaiteryhmään: tietokoneille, tableteille sekä älypuhelimille. (Cochran 2012, 2.) Nykyään Bootstrappia ei enää kehitä vain kaksi miestä, vaan sen kehittäjänä toimii kokonainen yhteisö, joka haluaa tarjota joukon parhaita, jaettavia käytäntöjä web-suunnittelijoille. Vaikka viralliselta kehittäjäyhteisöltä ei vaadita sen kummempaa kuuliaisuutta tai koodin yhdenmukaisuutta, Cochranin mukaan (2012, 2) yhteisölähtöinen kehitys kuitenkin toimii BS:ssä, sillä sen kautta tuotetaan aktiivisesti hyväksi havaittua ja testattua koodia.

Kolmas versio Bootstrapista julkaistiin elokuussa 2013. Siihen korjattiin huomattavasti aiemman version sisältämiä virheitä sekä tuotiin uusia ominaisuuksia. BS3:ssa oli otettu paremmin huomioon mobile first suunnittelu, jota silmällä pitäen esimerkiksi navigointipalkin toimintaa pienissä näyttötarkkuuksissa oli parannettu.

### 3.3 Käyttöluvat

Bootstrapin koodi on julkaistu MIT-lisenssin alla ja sen tekijänoikeudet kuuluvat Twitterille. MIT-lisenssi on vapaa ohjelmistolisenssi, joka sallii koodin käyttämisen myös

kaupallisissa projekteissa. Koodia saa vapaasti käyttää ja muokata sillä ehdolla, että lisenssin teksti säilyy lähdekoodissa. (Open Source Initiative.)

Käyttöehdot sisältävät myös tiettyjä rajoitteita, jotka määrittelevät miten tiedot lisenssistä ja tekijänoikeuksista tulee merkitä web-sivustolle sekä tekijöiden vastuunvapautuksen koodin käytöstä aiheutuvista haitoista (Bootstrap). Tarkat käyttöehdot ovat luettavissa Bootstrapin GitHub -projektissa (2014), license -tekstitiedostossa.

### **3.4 Bootstrap eri julkaisujärjestelmissä**

Bootstrap soveltuu kaikkiin yleisimpiin julkaisujärjestelmiin, joissa on mahdollisuus lisätä omat CSS ja JavaScript tiedostot web-sivustolle sekä muokata sivuston templaatteja. Sivuston templaatteihin pitää päästä käsin lisäämään BS:n grid-järjestelmä.

Olen käyttänyt Bootstrappia Wordpressissä, Joomlaassa, Drupalissa, Processwiressa sekä toimistomme omassa julkaisujärjestelmässä. Suosituimmat open source julkaisujärjestelmät, kuten WP, Joomla ja Drupal tarjoavat myös ilmaisia Bootstrap 3:seen pohjautuvia teemoja, joita voi jatkojalostaa omaan web-sivustoonsa sopivaksi. Valmiit teemat sisältävät normaalisti vaadittavat BS-tiedostot sekä oikealla tavalla rakennetun grid-järjestelmän. Valmiiden teemojen käyttäminen nopeuttaa tuotantoa ja säästää aikaa web-sivuston alustuksessa.

## 4 BOOTSTRAPIN KÄYTTÖ

Tällä hetkellä tuorein versio Bootstrapista on 3.1.1. Tässä osiossa läpikäytävät tekniikat ja tiedostomuodot viittaavat kyseiseen versioon ja sen puitteissa Bootstrapin web-sivustolla tarjottuun dokumentaatioon.

### 4.1 Tiedostojen latausmuodot

Bootstrapin käyttöoikeudet mahdollistavat sen latauksen ja käytön myös kaupallisissa projekteissa, jonka vuoksi web-suunnittelijat voivat käyttää sitä työkaluna rakentaessa myös laskutettavia web-sivustoja. Bootstrapin pohjätiedostot ovat ladattavissa neljällä eri tavalla.

#### 4.1.1 Source code

Source code eli lähdekoodi sisältää Bootstrapin alkuperäiset CSS, Less (CSS laajennus), JavaScript ja fontti tiedostot sekä dokumentaation. Kyseiset tiedostot löytyvät myös Bootstrapin GitHub:ista.

#### 4.1.2 Tuotantovalmis Bootstrap

Precompiled eli tuotantovalmis Bootstrap sisältää lähdekoodista käännettyt sekä pakatut versiot CSS ja JavaScript tiedostoista sekä fonttiedostot. Tämä muoto soveltuu kaikkiin HTML-pohjaisiin web-sivustoihin, työkaluista ja julkaisujärjestelmästä riippumatta. Se ei sisällä dokumentaatioita tai lisenssitietoja. Ladatut tiedostot puretaan web-sivuston juureen, josta ne linkitetään lokaalisti dokumentin head:ssa. Tuotantovalmiin Bootstrapin tiedostorakenne koostuu kolmesta kansioista: css, js ja fonts kansiot.

```

bootstrap/
├── css/
│   ├── bootstrap.css
│   ├── bootstrap.min.css
│   ├── bootstrap-theme.css
│   └── bootstrap-theme.min.css
├── js/
│   ├── bootstrap.js
│   └── bootstrap.min.js
├── fonts/
│   ├── glyphicons-halflings-regular.eot
│   ├── glyphicons-halflings-regular.svg
│   ├── glyphicons-halflings-regular.ttf
│   └── glyphicons-halflings-regular.woff

```

Pakatut tiedostot voi tunnistaa tiedostonimen `.min` päätteestä. Jos tarkoituksena ei ole päästä lukemaan tai muokkaamaan tiedostoja, Bootstrap suosittelee pakattujen tiedostojen käyttöä. Kyseiset tiedostot sisältävät saman koodin, kuin käännetyt tiedostot, mutta ne on tiivistetty, jolloin tiedostojen lataus palvelimelta vie vähemmän aikaa.

### 4.1.3 Sass

Sass:issa Bootstrapin Less tiedostot ovat käännetty Sass -muotoon (CSS laajennus). Sass on helppo lisätä Rails, Compass tai puhtaasti Sass -pohjaisiin web-sivustoihin.

### 4.1.4 Bootstrap CDN

Bootstrap CDN:ssä ei ole lainkaan ladattavia tiedostoja vaan CSS ja JavaScript tiedostot linkitetään projektiin ulkoiselta palvelimelta. MaxCDN tarjoaa Bootstrapille tuen, jolloin tiedostot linkitetään projektiin heidän palvelimiltaan. Alla esimerkki, miten Bootstrap tiedostot linkitetään web-sivuston head:ssa CDN:llä.

```

<!-- Uusin CSS -->
<link rel="stylesheet"
href="//netdna.bootstrapcdn.com/bootstrap/3.1.1/css/bootstrap.min.css">

<!-- Uusin JavaScript -->
<script
src="//netdna.bootstrapcdn.com/bootstrap/3.1.1/js/bootstrap.min.js"></script>

```

## 4.2 Selain- ja laitetuki

Bootstrap on suunniteltu toimimaan työpöytä- ja mobiiliselaimien viimeisimmillä versioilla, jonka vuoksi vanhemmilla selaimilla voi esiintyä eroavaisuuksia tyylien, toiminnallisuuksien ja tiettyjen komponenttien toiminnan kanssa. Windowsilla BS toimii Internet Explorer versioissa 8-11. Epävirallisesti BS toimii tarpeeksi hyvin myös Linuxin Chromium, Chrome ja Firefox -selaimilla, vaikka ne eivät ole virallisesti tuettuina. IE 7 versiolla grid-järjestelmä hajoaa, mutta muuten perustoiminnallisuudet ovat siinäkin epävirallisesti tuettu. (Bootstrap.)

TAULUKKO 5. Selain- ja laitetuki

Selain/käyttöjärjestelmä	Chrome	Firefox	IE 8-11	Opera	Safari
Android	kyllä	ei		ei	
iOS	kyllä			ei	kyllä
Mac OS X	kyllä	kyllä		kyllä	kyllä
Windows	kyllä	kyllä	kyllä	kyllä	ei

Tiettyjä virheitä koodista löytyy, jotka vaikuttavat sen toimintaan mm. vanhemmilla IE selaimilla. Yksi tällainen virhe on BS:n kuvakarusellin (carousel) liukuefektissä, joka on toteutettu CSS3 transform:lla. IE 9 ja sitä vanhemmat IE versiot eivät tue CSS3:n transform ominaisuutta, jonka vuoksi pitää tehdä muutoksia bootstrap.js tiedostoon. Bootstrap aiheilta keskustelupalstoilta löytyy korjattu JavaScript koodi, jolla kuvakarusellin liuku efektin saa toimimaan vanhemmilla IE versioilla.

Internet Explorer 8 ei itsessään tue media queriesia eikä HTML5:ttä, jonka vuoksi responsiivisuus ei toimi tällä selaimella automaattisesti. Responsiivisuuden saa kuitenkin lisättyä respond.js JavaScript lisäosalla. Bootstrap ei ole myöskään tuettu vanhemmissä IE:n yhteensopivuustiloissa, jonka vuoksi dokumentin head:iin tulee lisätä metatieto osoittamaan viimeisintä yhteensopivuustilaa.

```
<meta http-equiv="X-UA-Compatible" content="IE=edge">
```

```
<!--[if lt IE 9]>
  <script src="/js/html5shiv.js"></script>
  <script src="/js/respond.min.js"></script>
<![endif]-->
```



Yläpuolella olevassa esimerkissä luodaan ensin metatieto osoittamaan IE:n viimeisintä yhteensopivuustilaa. Sen jälkeen luodaan ehdollinen kommentti, jossa tarkistetaan onko IE:n selainversio vanhempi kuin IE 9. Jos versio on vanhempi, ladataan sivulle JavaScript tiedostot, jotka luovat selaimelle tuen HTML5 tekniikoille ja responsiivisen grid-järjestelmän. Huomioitavaa kuitenkin on, ettei responsiivisuus toimi IE 8:llä, jos sitä testataan lokaalisti `file://` kautta. Kun IE 8 selainta halutaan testata, tiedostot tulee siirtää palvelimelle.

Vaikka Bootstrap on monipuolisesti tuettu, markkinoiden kattavan laite- ja selaintarjonnan vuoksi sekin kaipaa välillä hieman hienosäätöä. Bootstrapin sivuilta löytyy kattava dokumentaatio tunnetuista laitekohtaisista korjauksista, josta voi ensisijaisesti hakea apua ongelmatilanteen kohdatessa.

### 4.3 Yleistä tekniikoista

Tässä osiossa käydään läpi Bootstrap 3.1.1. versiosta löytyviä front-end -tekniikoita, ja miten helposti erilaisilla määriteluokilla elementtien ominaisuuksia voidaan muuttaa. Määriteluokat tarkoittavat HTML elementeille annettavia lisäluokkia (`class`), joita voi olla useita samalla elementillä. BS:n määriteluokat tekevät ulkoasullisia ja toiminnallisia muutoksia HTML elementtiin tai sen sisältöön.

#### 4.3.1 Grid -järjestelmä

Bootstrap sisältää responsiivisen, ennalta määräytyvän grid-järjestelmän, joka on jaettu kahteentoista palstaan. Grid-järjestelmän avulla sivustolle luodaan asettelu, jossa sisältö jaetaan palstojen ja rivien sarjoihin.

Grid -järjestelmä ympäröidään `.container` (kiinteä leveys) tai `.container-fluid` (mukautuva leveys) `div` elementillä. Tämän jälkeen `.container`:in sisään laitetaan rivi (`row`).

```
<div class="container">
  <div class="row"></div>
</div>
```

Rivien (row) sisään rakennetaan sivuston palstat. Rivien tarkoitus on jakaa palstat horisontaalisiin ryhmiin. Sivuston sisältö sijoitetaan yksittäisten palstojen sisään.

```
<div class="container">
  <div class="row">
    <div class="col-md-6">Ensimmäinen palsta</div>
    <div class="col-md-6">Toinen palsta</div>
  </div>
</div>
```

Apuna rakenteen hahmottamisesta voi käyttää HTML taulukkoa, jossa rivi (tr) sisältää soluja (td). Taulukon sisältö, kuten teksti, grafiikka, ym., kirjoitetaan solun sisään. Yksittäisten solujen korkeus määrittää sitä ympäröivän rivin korkeuden. Grid-järjestelmässä palstat voi mieltää samankaltaisiksi kuin taulukon solu, sillä palstan sisään rakennetaan sivuston sisältö. Rivillä aloitetaan aina uusi palstaryhmä. Taulukosta poiketen grid-järjestelmän rivit voivat sisältää eri määrän palstoja, eli palstat eivät ole sarakkeissa, tai pystysuunnassa suhteessa toisiinsa.

```
<div class="container">
  <div class="row">
    <div class="col-md-12">Koko rivin levyinen palsta</div>
  </div>
  <div class="row">
    <div class="col-md-4">Ensimmäinen palsta</div>
    <div class="col-md-4">Toinen palsta</div>
    <div class="col-md-4">Kolmas palsta</div>
  </div>
</div>
```

Palstojen välistys on luotu palstatäytteellä (padding), joka on oletuksena oikeasta ja vasemmasta laidasta 15 pikseliä. Koska palstatäyte on molemmilla reunoilla, rivin ensimmäisen ja viimeisen palstan sisältöalue .container:iin nähden ei ole linjassa. Rivien vasempaan ja oikeaan reunaan on luotu palstoja varten kohdistus negatiivisella margi-

naalilla -15 pikseliä. Tämä muuttaa rivin ensimmäisen ja viimeisen palstan sijaintia niin, että niiden sisältöalue on `.container:`in kanssa samassa linjassa.



Kuva 3. Esimerkki grid-järjestelmästä, jossa on merkitty palstatäyte ja negatiivinen marginaali

Palstojen leveys on tyylitiedostossa prosentteina ja palstat luodaan antamalla niille CSS määriteluokkaan numero 1-12 väliltä. Koko rivin levyisen palstan luokka on `.col-md-12` ja esimerkiksi rivillä, jossa on kolme saman levyistä palstaa, luokat ovat `.col-md-4`.

Palstan määriteluokilla voidaan vaikuttaa, minkä levyinen se on eri näyttötarkkuuksissa. Bootstrapissa on yhteensä neljä valmiiksi luotua breakpoints määritelmää palstoille: `lg`, `md`, `sm` ja `xs`.

TAULUKKO 6. Breakpoints

Luokka	Leveys pikseleinä	Kohdistettu
<code>.col-lg-*</code>	> 1200	suuret työpöydät
<code>.col-md-*</code>	> 992	pienet työpöydät
<code>.col-sm-*</code>	> 768	tabletit ja pienet näytöt
<code>.col-xs-*</code>	< 768	älypuhelimet

Sivusto voi esimerkiksi olla jaettuna neljään palstaan, kun sitä katsellaan työpöydällä. Tabletinäkymässä palstat ovat jaettu kahteen riviin ja palstoja on rinnakkain vain kaksi. Mobiilissa palstat ovat aina 100 prosenttia leveät, ellei niille määritellä erillistä `.col-xs-*` -luokkaa.

```

<div class="container">
  <div class="row">
    <div class="col-sm-6 col-md-3">Palsta 1</div>
    <div class="col-sm-6 col-md-3">Palsta 2</div>
    <div class="col-sm-6 col-md-3">Palsta 3</div>
    <div class="col-sm-6 col-md-3">Palsta 4</div>
  </div>
</div>

```



KUVA 4. Palstojen mukautuminen erikokoisille näytöille.

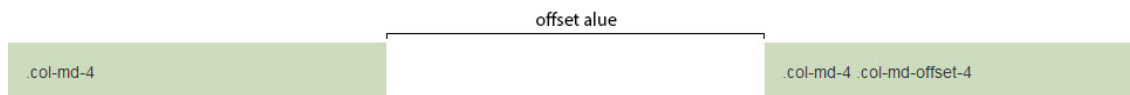
#### 4.3.1.1 Kohdistetut palstat

Palstalle, joka ei ole koko rivin levyinen, voidaan luoda \* palstan levyinen kohdistus käyttämällä määriteluokkaa `.col-md-offset-*`. Jos esimerkiksi `.col-md-4` palstalle halutaan luoda neljän palstan levyinen tyhjä tila vasempaan reunaan, lisätään palstan luokaksi `.col-md-offset-4`.

```

<div class="row">
  <div class="col-md-4">.col-md-4</div>
  <div class="col-md-4 col-md-offset-4">.col-md-4 .col-md-offset-4</div>
</div>

```



KUVA 5. Kohdistetun palstan offset alue

### 4.3.1.2 Palstajärjestys

Palstojen järjestystä voidaan muuttaa käyttämällä palstassa määriteluokkia `.col-md-push-*` ja `.col-md-pull-*`. Esimerkiksi työpöytänäytölle vasempaan reunaan määritelty palsta voi sijaita oikeassa reunassa tablettinäytöllä.

```
<div class="row">
  <div class="col-sm-9 col-md-9 col-md-push-3">.col-md-9</div>
  <div class="col-sm-3 col-md-3 col-md-pull-9">.col-md-3</div>
</div>
```

Työpöytänäyttö



Tabletti



KUVA 6. Palstajärjestys työpöytä- ja tablettinäytöllä

## 4.3.2 Typografia

Suurin osa mobile first suunnittelusta koostuu typografiasta. Se on monipuolinen tapa luoda myös pieniin näyttötarkkuuksiin visuaalisuutta, kasvattamatta sivuston latausai-kaa. Tämä on otettu huomioon myös Bootstrapin tyylessä, joiden avulla voi helposti rakentaa typografisesti monipuolisia sivustoja.

Bootstrapista löytyy valmiiksi määritellyt tyylit erilaisille typografisille elementeille, kuten otsikoille, linkeille ja leipätekstille. Otsikoissa tuettuina ovat otsikkoelementit `h1-h6` sekä niistä luodut määriteluokat `.h1-.h6`. Otsikkojen määriteluokkien avulla eri otsikkokokoja voidaan käyttää yhdessä ilman rivinvaihtoa. Otsikoissa voidaan lisäksi hyödyntää `<small>` elementtiä tai `.small` luokkaa, mikäli otsikolle halutaan luoda tois-sijainen teksti.

**h1. Bootstrap otsikko** Toissijainen teksti

**h2. Bootstrap otsikko** Toissijainen teksti

**h3. Bootstrap otsikko** Toissijainen teksti

**h4. Bootstrap otsikko** Toissijainen teksti

**h5. Bootstrap otsikko** Toissijainen teksti

**h6. Bootstrap otsikko** Toissijainen teksti

KUVA 7. Otsikkotyylit

Bootstrap määrittelee yleisesti teksteille oletusfonttikooksi 14 pikseliä (`font-size`) ja rivinkorkeudeksi 1.482 (`line-height`). Leipätekstin (`p`) alamarginaali on laskettu rivin korkeudesta, joka on oletuksena 10 pikseliä (`margin-bottom`). Leipätekstille voi antaa erilaisia määriteluokkia, kuten `.lead` tai `.small`, jotka muuttavat tekstin kokoa sen käyttötarkoituksen mukaan.

Yleisesti kaikkien tekstien tasausta voi muuttaa määriteluokkien avulla. Tekstielementtien tasauksen määriteluokat ovat vasemmalle (`.text-left`), keskelle (`.text-center`), oikealle (`.text-right`) ja tasattu (`.text-justify`).

### 4.3.3 Painikkeet

Erilaisilla painikkeilla käyttäjää voidaan ohjata menemään sivustolla tiettyyn paikkaan. Painikkeiden tyylittelyn tulee kertoa käyttäjälle että kyseinen elementti on klikattava ja sillä on erityistä huomioarvoa sivustolla. Bootstrap tarjoaa valmiit tyylittelyt painikkeille eri kokoihin ja käyttötarkoituksiin.

Painikkeiden ulkoasulla voidaan käyttäjälle helposti kertoa, mitä sitä painettaessa tapahtuu. BS:ssä on painikkeille seitsemän erilaista määriteluokkaa, kuvastamaan eri käyttötarkoituksia.

```
<button type="button" class="btn btn-default">Default</button>
<button type="button" class="btn btn-primary">Primary</button>
<button type="button" class="btn btn-success">Success</button>
```

```

<button type="button" class="btn btn-info">Info</button>
<button type="button" class="btn btn-warning">Warning</button>
<button type="button" class="btn btn-danger">Danger</button>
<button type="button" class="btn btn-link">Link</button>

```



KUVA 8. Painikkeiden ulkoasu

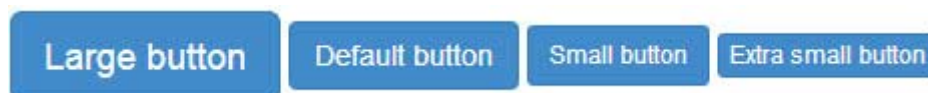
Painikkeiden määriteluokkia voidaan käyttää esim. a, button, ja input elementeissä. Bootstrapin dokumentaatioissa suositellaan, että painikkeina käytettäisiin aina ensisijaisesti button elementtiä, jotta ne toistuvat yhtenäisesti eri selaimissa. Määriteluokka ei muuta elementin toiminnallisuutta vaan yksinkertaisesti lisää sille erottuvan CSS tyylin.

Painikkeiden huomioarvoa voi myös korostaa käyttämällä napin kokoon vaikuttavia määriteluokkia. Bootstrap tarjoaa painikkeista neljää eri kokoa: oletuskoko (default), suuri (.btn-lg), keskikokoinen (.btn-sm) ja pieni (.btn-xs).

```

<button type="button" class="btn btn-primary btn-lg">Large button</button>
<button type="button" class="btn btn-primary">Default button</button>
<button type="button" class="btn btn-primary btn-sm">Small button</button>
<button type="button" class="btn btn-primary btn-xs">Extra small button</button>

```



KUVA 9. Painikkeiden koot

#### 4.3.4 Kuvat

Koska mukautuvat kuvat ovat hyvin oleellinen osa responsiivista suunnittelua, Bootstrapista löytyy määriteluokkia responsiivisten kuvien luontiin sekä kuvien erilaisiin käyttötarkoituksiin.

Kuvalle lisätään määriteluokka .img-responsive, jolloin sen maksimileveys on 100 % ja korkeus automaattinen. Korkeuden määrittäminen automaattiseksi estää kuvasuhteen hajoamisen leveyttä pienemmäksi skaalattaessa, vaikka kuvalle olisi HTML-koodissa

annettu `width` ja `height` arvot. Kuva skaalautuu sen isäntäelementtiin nähden, eli jos kuva on palstassa, skaalautuminen määräytyy palstan leveyden mukaisesti.

Kun halutaan, että kaikki sivuston kuvat ovat automaattisesti responsiivisia, ilman erillisen määriteluokan käyttämistä kuvassa, sen voi luoda käsin omassa `custom.css` tiedostossa. CSS tiedostossa kohdistetaan tyyli `img` elementtiin ja annetaan sille maksimileveydeksi 100 prosenttia sekä automaattinen korkeus. Tällöin tyyli vaikuttaa kaikkiin sivustolta löytyviin `img` elementteihin.

```
img{
  max-width: 100%;
  height: auto;
}
```

Bootstrap tarjoaa kuvalle myös sen ulkoasuun vaikuttavia määriteluokkia. Kuvalle voi luoda pyöristetyt kulmat, se voi olla kokonaan pyöreä tai esikatselukuva.

```



```



KUVA 10. Kuvan ulkoasut

### 4.3.5 Lomakkeet

Bootstrap sisältää valmiit tyylit kaikille yleisimmille lomake-elementeille, kuten `input`, `select`, `textarea`, `checkbox` ja niin edelleen.



### 4.3.5.1 Peruslomake

Peruslomakkeen elementit luodaan `.form-group` `div`:n sisään, jolloin niistä tulee responsiivisia, oletusleveydeltään 100 prosenttia isäntäelementtiin nähden. Peruslomakkeessa kaikki elementit ovat kasattu allekkain.

```
<form role="form">
  <div class="form-group">
    <label for="exampleInputEmail1">Sähköposti</label>
    <input type="email" class="form-control" id="exampleInputEmail1"
  </div>
  <div class="form-group">
    <label for="exampleInputPassword1">Salasana</label>
    <input type="password" class="form-control" id="exampleInputPassword1">
  </div>
  <div class="checkbox">
    <label>
      <input type="checkbox"> Hyväksyn
    </label>
  </div>
  <button type="submit" class="btn btn-default">Lähetä</button>
</form>
```

**Sähköposti**

**Salasana**

Hyväksyn

Lähetä

KUVA 11. Peruslomake

### 4.3.5.2 Inline lomake

Lomakkeesta voi luoda myös yhdelle riville sijoittuvan version, inline lomakkeen, jolloin se vie vähän tilaa korkeussuunnassa. Inline lomakkeen saa aktivoitua lisäämällä form elementille määriteluokan `.form-inline`.

Inline lomakkeissa ei normaalisti näytetä nimiöitä (label) esimerkiksi tekstikentän edessä kertomaan, mitä tietoa siihen tulisi syöttää, vaan se hoidetaan placeholderilla. Nimiöitä ei silti saa missään nimessä poistaa lomakkeesta, sillä muuten näytönluohjelmat eivät osaa kertoa, mikä tekstikenttä on kyseessä. Nimiö elementeille annetaan määriteluokka `.sr-only`, joka näyttää nimiön vain näytönluohjelmille.

```
<form class="form-inline" role="form">
  <div class="form-group">
    <label class="sr-only" for="exampleInputEmail2">Sähköposti</label>
    <input type="email" class="form-control" id="exampleInputEmail2"
      placeholder="Sähköposti">
  </div>
  <div class="form-group">
    <label class="sr-only" for="exampleInputPassword2">Salasana</label>
    <input type="password" class="form-control" id="exampleInputPassword2"
      placeholder="Salasana">
  </div>
  <div class="checkbox">
    <label>
      <input type="checkbox"> Hyväksyn
    </label>
  </div>
  <button type="submit" class="btn btn-default">Lähetä</button>
</form>
```



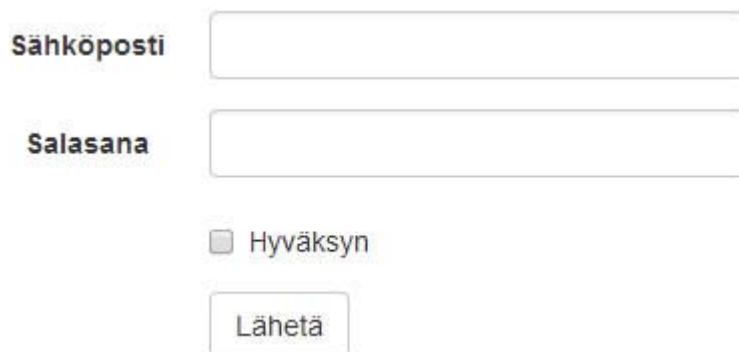
The image shows a rendered version of the HTML code above. It consists of three input fields arranged horizontally. The first is a text input with the placeholder 'Sähköposti'. The second is a password input with the placeholder 'Salasana'. The third is a checkbox with the label 'Hyväksyn'. To the right of the checkbox is a 'Lähetä' (Submit) button.

KUVA 12. Yhdelle riville sijoittuva inline -lomake

### 4.3.5.3 Horisontaalinen lomake

Horisontaalisessa lomakkeessa esim. nimiöt ja tekstikentät voidaan sijoittaa samaan riviin käyttämällä Bootstrapin grid-järjestelmää. Horisontaalisen lomakkeen saa aktivoitua lisäämällä `form` elementille määriteluokan `.form-horizontal`. Tämän jälkeen lomakkeen sisälle voi luoda oman grid-järjestelmän. Grid-järjestelmä toimii samalla tavalla kuin sivuston palstoitusta tehtäessä, mutta lomakkeessa rivit merkitään `.form-group` luokalla, joten erillisiä `.row` luokkia ei tarvita.

```
<form class="form-horizontal" role="form">
  <div class="form-group">
    <label for="inputEmail3" class="col-md-3 control-label">Email</label>
    <div class="col-md-9">
      <input type="email" class="form-control" id="inputEmail3">
    </div>
  </div>
  <div class="form-group">
    ...
  </div>
  <div class="form-group">
    <div class="col-md-offset-3 col-md-9">
      <button type="submit" class="btn btn-default">Sign in</button>
    </div>
  </div>
</form>
```



Screenshot of a horizontal Bootstrap form. The form consists of three rows. The first row has a label 'Sähköposti' followed by a text input field. The second row has a label 'Salasana' followed by a text input field. The third row has a checkbox labeled 'Hyväksyn'. Below the checkbox is a button labeled 'Lähetä'.

KUVA 13. Horisontaalinen lomake

## 4.3.6 Taulukot

Taulukon oletusmuotoilu on HTML:ssä melko karu ja siksi Bootstrap tarjoaa taulukolle muutamia perusmuotoiluja, joilla sen saa helposti modernin näköiseksi sekä halutessaan responsiiviseksi. BS tyyllittelyt saa lisättyä taulukon `table` elementtiin määriteluokkia hyödyntämällä.

### 4.3.6.1 Perusmuotoilut

Perustaulukolle riittää määriteluokka `.table`, jolloin taulukon soluille on annettu yläosaan yhden pikselin paksuinen reunus. Tämä jakaa solut selkeästi omiin riveihinsä, eikä muita reunuksia taulukossa ei ole. Määriteluokkaa `.table` käytetään pohjana kaikissa BS taulukkomuotoiluissa.

Taulukkoon voi lisätä seepra-efektin, missä joka toisella rivillä on eri taustaväri. Seepra-tilin saa käyttöön lisäämällä taulukolle määriteluokan `.table-striped`. Kehystetyssä taulukossa soluilla on joka laidalla reunus. Kehystetyn taulukon määriteluokka on `.table-bordered`.

```
<table class="table">
  <thead>...</thead>
  <tbody>...</tbody>
</table>

<table class="table table-striped">
  <thead>...</thead>
  <tbody>...</tbody>
</table>

<table class="table table-bordered">
  <thead>...</thead>
  <tbody>...</tbody>
</table>
```

Perustaulukko		Seeprataulukko		Kehystetty taulukko	
Sija	Nimi	Sija	Nimi	Sija	Nimi
1	Etunimi Sukunimi	1	Etunimi Sukunimi	1	Etunimi Sukunimi
2	Etunimi Sukunimi	2	Etunimi Sukunimi	2	Etunimi Sukunimi
3	Etunimi Sukunimi	3	Etunimi Sukunimi	3	Etunimi Sukunimi

KUVA 14. Taulukkomuotoilut

#### 4.3.6.2 Hover-efekti

Bootstrapista löytyy taulukolle tarkoitettu määriteluokka, jolla käyttäjälle tarjotaan helpotusta pitkän taulukon lukemiseen. Hover-efektissä taulukon rivin taustaväri muuttuu, kun siihen viedään hiiri päälle. Hover-efektin saa käyttöön lisäämällä taulukolle määriteluokan `.table-hover`.

#### 4.3.6.3 Responsiivinen taulukko

Taulukoiden lukeminen mobiililaitteella on hankalaa, mikäli niille ei ole tehty minkäänlaista responsiivista muotoilua. Taulukon saa responsiiviseksi luomalla sen `.table-responsive` div elementin sisään. Responsiivisessa taulukossa `.table-responsive` div on 100 prosenttia leveä isäntäelementtiinsä nähden ja sen sisältö on horisontaalisesti vieritettävissä.

```
<div class="table-responsive">
  <table class="table table-striped">
    <thead>...</thead>
    <tbody>...</tbody>
  </table>
</div>
```

Sija	Nimi	Lisätiedot
1	Etunimi Sukunimi	Lorem ipsum dolor
2	Etunimi Sukunimi	Lorem ipsum dolor
3	Etunimi Sukunimi	Lorem ipsum dolor

KUVA 15. Responsiivinen taulukko pienessä näyttöleveydessä

#### 4.3.6.4 Rivin asiayhteys

Taulukon riveille voi myös antaa erilaisia määriteluokkia, joilla rivin voi visuaalisesti yhdistää tiettyyn asiayhteyteen. Määriteluokkia on viisi erilaista: aktiivinen (`.active`), onnistunut (`.success`), info (`.info`), varoitus (`.warning`) ja vaara (`.danger`).

#### 4.3.7 Vaihtoehdoisen sisällön näyttäminen

Kun sivustosta rakennetaan responsiivista, tulee ottaa huomioon, ettei kaikkea sisältöä ole ehkä järkevä näyttää käyttäjälle pienissä mobiilinäyttöissä. Bootstrapin responsiivisilla apuvälineillä (`responsive utilities`) voidaan määrittää, mitä elementtejä käyttäjä näkee erikokoisilla näyttölaitteilla. Vaihtoehdoisen sisällön näyttäminen perustuu `media queries`in avulla rakennettuihin määriteluokkiin.

Alla olevassa taulukossa näyttöjen lyhenteet vastaavat seuraavia:

XS = Pieni mobiililaitte (<768 px)

SM = Tabletti (>768 px)

MD = Pieni työpöytä (>992 px)

LG = Suuri työpöytä (>1200 px)

TAULUKKO 7. Vaihtoehdoisen sisällön määriteluokat

	XS näyttö	SM näyttö	MD näyttö	LG näyttö
.visible-xs	näkyvissä			näkyvissä
.visible-sm		näkyvissä		
.visible-md			näkyvissä	
.visible-lg				näkyvissä
.hidden-xs		näkyvissä	näkyvissä	näkyvissä
.hidden-sm	näkyvissä		näkyvissä	näkyvissä
.hidden-md	näkyvissä	näkyvissä		näkyvissä
.hidden-lg	näkyvissä	näkyvissä	näkyvissä	

Vaihtoehdoisen sisällön näyttämällä voidaan myös määrätä, mitä elementtejä sivua tulostaessa näytetään. Tulostuksen määriteluokat ovat `.visible-print` ja `.hidden-print`.

#### 4.4 Navigaatiot

Bootstrap tarjoaa valmiit rakenteet erilaisten navigaatioiden luomiseen. Kaikki BS:n navigaatiot pohjautuvat järjestelemättömiin listoihin (unordered lists) ja erilaisia navigaatiotyyppettä yhdistää `ul.nav` luokka. Alla oleva esimerkki muotoilemattoman navigaation rakenteesta ei sisällä määriteluokkia.

```
<ul class="nav">
  <li><a href="#">Etusivu</a></li>
  <li><a href="#">Tuotteet</a></li>
  <li><a href="#">Yhteytiedot</a></li>
</ul>
```

##### 4.4.1 Navigaation määriteluokat

Navigaatiotyyppettä saa muutettua antamalla `ul` elementille erilaisia määriteluokkia. Välilehtinavigaation saa määriteltä lisäämällä `ul` elementille määriteluokan `.nav-tabs`. Vastaavalla tavalla rakennetaan myös pillerinavigaatio, jonka määriteluokka on `.nav-`

`pills`. Kolmas määriteluokka, `.navbar-nav`, on käytössä vain navigointipalkin sisään luotavissa navigaatioissa.

```
<ul class="nav nav-tabs">
```

```
...
```

```
</ul>
```

```
<ul class="nav nav-pills">
```

```
...
```

```
</ul>
```

```
<ul class="nav navbar-nav ">
```

```
...
```

```
</ul>
```



KUVA 16. Välilehti- ja pillerinavigaatio

Navigaation määriteluokilla voi myös helposti muuttaa onko navigaatio tasattu isäntä-elementtinsä reunoihin (`justified`), jolloin kaikki `<li>`-elementit ovat yhtä leveitä. Pienimmillä, alle 768 pikselin levyisillä näytöillä `<li>`-elementit kasaantuvat allekkain.

```
<ul class="nav nav-pills nav-justified">
```

```
...
```

```
</ul>
```



KUVA 17. Tasattu pillerinavigaatio

Pillerinavigaation voi myös määrätä kasaantuvaksi allekkain myös isommissa näyttötarkkuuksissa, lisäämällä `ul` elementille toisen määriteluokan `.nav-stacked`.

```
<ul class="nav nav-pills nav-stacked">
```

```
...
```

```
</ul>
```





KUVA 18. Vertikaalinen pillerinavigaatio.

#### 4.4.2 Navigointipalkki

Navigointipalkki (navbar) esiteltiin BS2 versiossa. Tämä täysin responsiivinen navigaatio suunniteltiin pieneen laitteeseen mukautuvaksi. Suuremmilla näyttölaitteilla, eli yli 768 pikseliä leveillä näytöillä, navigaatio esitetään horisontaalisesti. Navigointipalkki sisältää romahtavan navigaation (collapse nav), joka tarjoaa alle 769 pikseliä leveälle näytölle, eli mobiilikäyttäjälle painikkeen takaa löytyvän listauksen sivuista (Cochran 2012, 30). Tämä tapa esittää navigaatio pienillä näyttölaitteilla on melko hyvin vakiinnuttanut asemansa, eivätkä Bootstrapin kehittäjät ole nähneet tarpeelliseksi muuttaa sitä BS3 versioon.

Navigaatiopalkki on rakenteeltaan melko kattava. Sen perusrunko alkaa aina `.navbar` luokalla ja navigaation voi rakentaa joko `div` tai `nav` elementtiin.

```
<nav class="navbar navbar-static-top" role="navigation">
  ...
</nav>
```

Navigaatiolle voi lisätä erilaisia määriteluokkia, kuten normaaliin navigaatioonkin. Määriteluokilla voidaan antaa navigaatiopalkille esimerkiksi tietty sijainti. Sijaintimääriteluokkia on kahta eri tyyppiä: staattinen (`.navbar-static-top`) ja kiinnitetty (`.navbar-fixed-top` sekä `.navbar-fixed-bottom`). Staattinen navigaatio pysyy aina siinä kohdassa sivustoa mihin se on määritelty. Kiinnitetyn navigaation sijainti on taas suhteellinen ruutuun nähden, eli se pystyy aina tiettyssä kohtaa ruutua, riippumatta siitä, kuinka alas käyttäjä on sivua vierittänyt. Navigaatiopalkin värimaailman voi nopeasti muuttaa käänteiseksi lisäämällä sille määriteluokan `.navbar-inverse`.

```

<nav class="navbar navbar-static-top" role="navigation">
  <div class="container">
    ...
  </div><!-- /.container -->
</nav>

```

Navigation sisään määritetään `.container` tai `.container-fluid` osoittamaan sen sisältöelementtien kokonaisleveyttä.

```

<nav class="navbar navbar-static-top" role="navigation">
  <div class="container">
    <div class="navbar-header">
      <button type="button" class="navbar-toggle" data-toggle="collapse"
        data-target="#collapse-1">
        <span class="sr-only">Toggle navigation</span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
      </button>
      <a class="navbar-brand" href="#">Yritys Oy</a>
    </div>
  </div><!-- /.container -->
</nav>

```

Navigation `.nav-header` div:n sisään määritellään sivuston nimi tai logo sekä romahtavan navigation vaihtopainike (`.navbar-toggle`). Vaihtopainikkeen `.navbar-toggle` -luokan avulla BS:n tyylessä määritetään, että elementti näytetään vain alle 768 pikseliä leveillä näytöillä. Vaihtopainikkeelle luodaan romahtavan navigation avaus- ja sulke- mistöiminnallisuus antamalla sille `data-toggle="collapse"` ja kohteeksi romahtettavan div elementin ID `data-target="#collapse-1"`. ID on aina yksilöllinen, eli samaa ID arvoa ei voi käyttää kuin yhdessä div elementissä. Vaihtopainikkeen sisään voi määrittellä viiva-ikonin tai vaihtoehtoisesti kirjoittaa tekstiä painikkeen sisään, esimerkiksi MENU.

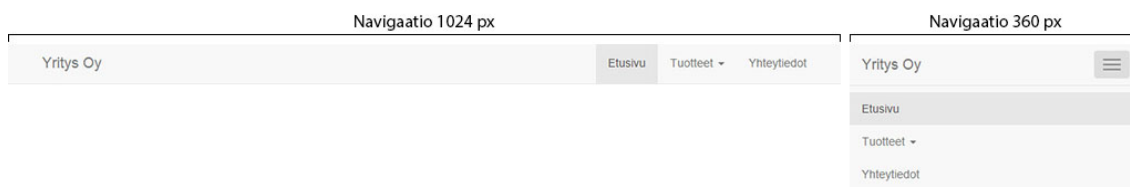
```

<nav class="navbar navbar-default navbar-static-top" role="navigation">
  <div class="container">
    <div class="navbar-header">
      ...
    </div>
    <div class="collapse navbar-collapse" id="collapse-1">
      <ul class="nav navbar-nav navbar-right">
        ...
      </ul>
    </div><!-- /.navbar-collapse -->
  </div><!-- /.container -->
</nav>

```

Romahtava navigaationlistaus ympäröidään div elementillä, jolle annetaan luokat `.collapse` ja `.navbar-collapse`. Tämän div:n ID tulee olla sama, kuin mikä määriteltiin vaihtopainikkeen `data-target` elementtiin.

Itse sivulistaukselle voidaan erikseen määrätä, onko se tasattu oikeaan vai vasempaan reunaan, antamalla `ul.nav` elementille luokat `.nav-right` tai `.nav-left`.



KUVA 19. Navigaatio 1024 pikseliä ja 360 pikseliä leveillä näytöillä.

## 4.5 Komponentit

Bootstrap tarjoaa kattavan valikoiman erilaisia komponentteja, joilla web-sivustolle voi rakentaa ikoneita, pudotusvalikoita, murupolkuja, alert -viestejä ja paljon muuta. Komponentit aktivoidaan antamalla elementille jokin tietty rakenne ja määriteluokka. Useat komponentit vaativat toimiakseen sille tarkoitetun jQuery lisäosan. Komponentit tarjoavat web-sivustolle perussetin erilaisia käyttöliittymäelementtejä.

## 4.6 jQuery lisäosat

Bootstrapin komponentit saadaan toimimaan lataamalla web-sivustolle jQuery lisäosia. Lisäosien avulla erilaisiin komponentteihin luodaan niiden toiminnallisuus. Lisäosilla saadaan esimerkiksi alert -viestin sulkemispainikkeelle tekniikkaa, millä viesti-ikkuna oikeasti menee piiloon tai pudotusvalikolle tekniikka sen avaamisen ja sulkemiseen.

## 4.7 Kustoimointi web-projektiin

Tässä osiossa käsitellään millaisilla eri tavoilla Bootstrapin valmiita tyylejä voi mukauttaa omaan projektiin sopivaksi, jotta web-sivuston ulkoasuun saa muutettua toivotuksia.

### 4.7.1 Kustomoitujen tiedostojen lataaminen

Bootstrap tarjoaa sivustollaan vaihtoehdon, jossa web-sivustoa varten voi ladata mukautetut CSS tiedostot, Less-muuttujat, komponentit sekä jQuery lisäosat. Mikäli web-sivusto on pienibudjettinen eikä vaadi kovin monimutkaisia toiminnallisuuksia, on järkevää jättää tiedostoista pois ominaisuudet, joita ei tarvitse. Tämä pienentää tiedostokoja ja nopeuttaa web-sivuston latausaikaa.

Bootstrapin sivuilla löytyy Customize osio, josta suunnittelija voi valita itse sivustoonsa sopivimmat osat. Ajatellaan esimerkkinä kampanjasivustoa, joka on julkinen vain tietyn aikaa. Kampanjassa järjestetään arvonta, jonka tarkoituksena on kerätä osallistujilta yhteystietoja jatkomarkkinointia varten. Sivusto sisältää responsiivisen taiton, tekstiä ja osallistumislomakkeen. Näin suppeaan web-sivustoon tarvitaan vain murto-osa BS:n tarjoamista ominaisuuksista ja tekniikoista. Kampanjan suunnittelija miettii etukäteen millaisia toiminnallisuuksia sivustolla tulee olemaan ja rastittaa vain nämä vaihtoehdot Customize osiossa ladattaviksi Bootstraptiin.

Mikäli sivustolle halutaan määrittää jo taiton alustusvaiheessa BS:n oletuksesta poikkeavia tyylejä, kuten kampanjan värimaailma, fonteja tai nappien värit, ne voi muuttaa Less-muuttujien listaukseen. Less-muuttujat toimivat samalla tavoin kuin peruskoodauksesta tutut muuttujat (variables). Less-muuttujille voidaan määrittää tietty arvo, jota

kutsutaan myöhemmin tyylitiedoston eri kohdissa. Kampanjasivuston grid-järjestelmän ja breakpointsit voi myös helposti mukauttaa omaan tarpeeseen, mikäli ei ole tyytyväinen miten web-sivusto näkyy tietyssä näyttökoossa tai vaatimuksena on muuttaa palstatäytteen leveyttä.

Jos web-sivustolle on määritelty selkeät vaatimukset, on viisasta että suunnittelija käyttää ennen taiton aloittamista hetken mukautettujen tiedostojen laatimiseen, sillä nämä vähentävät huomattavasti jälkikäteen tehtävää tyylien ylikirjoittamista.

#### 4.7.2 Bootstrapin tyylien ylikirjoittaminen

Yleinen tapa luoda BS pohjaisesta web-sivustosta yksilöllinen on ylikirjoittaa sen tarjoamia valmiita CSS tyyliä. Koskaan ei kannata lähteä tekemään muutoksia bootstrap.css tai bootstrap.min.css tiedostoon, jotta BS pysyy päivitettävänä. Ylikirjoitusta varten luodaan oma tyylitiedosto. Oma CSS tiedosto ladataan sivustolle vasta bootstrap.min.css tiedoston jälkeen, jotta ylikirjoittaminen on mahdollista.

```
<link rel="stylesheet" href="/css/bootstrap.min.css" media="all" />
<link rel="stylesheet" href="/css/custom.css" media="all" />
```

Ylikirjoittamisen periaate on se, että samalle elementille määritellään kaksi erilaista tyyliä, joista myöhemmin ladatun ominaisuudet ylikirjoittavat edellisen. Esimerkiksi haluamme muuttaa BS:n sivuston oletustekstin väriä, joka on BS:n omassa tyylitiedostossa #333 värinen, eli tumma harmaa. Oletustekstin värin määritellään CSS tyylitiedoston alussa body elementille, josta se vaikuttaa kaikkiin sivuston teksteihin, joille ei ole erikseen määritelty väriä.

##### **bootstrap.min.css**

```
body{ color: #333; }
```

##### **custom.css**

```
body{ color: #BADA55; }
```

Kun body:lle annetaan uusi väri #BADA55 BS:n tyylien jälkeen ladattavassa custom.css -tiedostossa, sivusto muuttaa oletustekstin värin vihreäksi. Samalla periaatteella Bootst-

rapin oletustyyleistä voi ylikirjoittaa CSS:llä kaiken haluamansa ja tehdä näin web-sivustosta persoonallisen näköisen.

## 5 BOOTSTRAP - SWOT ANALYYSI

Bootstrap on framework, josta löytyy omat vahvuudet ja heikkoudet. Tässä kappaleessa käydään läpi, mitkä asiat puoltavat BS:n käyttöä web-sivustossa ja mitkä asiat tulisi ottaa huomioon kyseistä frameworkkia valitessa.

### 5.1 Vahvuudet ja mahdollisuudet

Bootstrapin tekniikassa on paljon vahvuuksia, jotka huomattavasti vähentävät käsin tehtävän työn määrää ja näin kasvattavat web-sivuston kustannustehokkuutta. Laajan kehittäjäyhteisönsä ansiosta BS:ään julkaistaan säännöllisesti päivityksiä, joissa on korjattu käyttäjien toimesta ilmoitettuja virheitä. Laadukas, hyvin kommentoitu lähdekoodi on kattavasti dokumentoitu, joka helpottaa tekniikoiden omaksumista. Yhdenmukaisen koodin ansiosta web-sivuston päivittäminen jälkikäteen on helpompaa, vaikka päivittäjä olisi eri henkilö, kuin itse sivuston suunnittelija tai taittaja.

Bootstrap on tukee kaikkia moderneja web-selaimia, joka vähentää testaukseen käytettävää aikaa. Vaikka suunnittelija törmäisi ongelmiin jonkin tietyn selainversion tai päätelaitteen kanssa, voi virallisesta dokumentaatiosta etsiä ratkaisua. Erikoisempiin ongelmiin löytää apua myös virallisen dokumentaation ulkopuolelta. Monilla web-kehittäjien foorumeilla on oma keskusteluketju BS:n tekniikoita varten, jossa ratkotaan ongelmia aktiivisesti.

Bootstrap on helppo kustomoida omaan projektiin sopivaksi ja näin luoda persoonallinen ulkoasu web-sivustolle. Vaikka BS tarjoaa laajan valikoiman erilaisia tekniikoita, mukautetun latauksen avulla suunnittelija voi luoda Bootstrapista version, joka soveltuu juuri hänen web-sivustoonsa. Lisäksi BS:n valmiita tyylejä on mahdollista ylikirjoittaa. Vaikka BS ei muutenkaan ole massiivisen kokoinen framework, voi kustomoinnilla lyhentää web-sivuston latausaikaa.

Responsiivinen, kahdentoista palstan grid-järjestelmä mukautuu automaattisesti kaikkiin näyttötarkkuuksiin, jolloin sivuston saa automaattisesti toimimaan niin älypuhelimella kuin SmartTV:llä. Palstojen leveyksiä on helppo hallita määriteluokilla, jonka vuoksi eri näyttökokoihin mukautuvat näkymät on nopea luoda.

Bootstrap tarjoaa jo melko kattavasti kolmannen osapuolen ominaisuuksia, jonka vuoksi ominaisuuksien laajentuminen tulevaisuudessa on todennäköistä. Internetistä hakemalla löytyy myös hyvin kattavasti käyttäjien tekemiä BS:ään pohjautuvia epävirallisia teemoja sekä lisäosia, joiden avulla BS:n frameworkkia voi laajentaa omiin tarpeisiinsa sopiviksi.

Web-sivuston suunnittelun aloitus jo ulkoasun suunnitteluvaiheessa on aktiivisten käyttäjien ansioista helppoa: useilla eri sivustoilla on tarjolla ilmaiseksi ladattavia Adobe Photoshop pohjia, joihin on rakennettu BS:n grid-järjestelmä pohjalle. Pohjia löytyy eri näyttötarkkuuksiin, työpöytänäytöstä iPhoneen näyttöön. Lisäksi ladattavana on valmiiksi piirretyt BS:n käyttöliittymäelementit, kuten painikkeet, ikonit, lomakekentät, jne. Valmiiden elementtien avulla suunnittelijan ei tarvitse käyttää aikaa piirtopöydällä niiden luomiseen, vaan hän voi vain asetella valmiit elementit vedokseen ja mahdollisesti muuttaa hieman värejä. Kun esim. painikkeet ovat jo vedoksessa oikean muotoisia ja kokoisia, web-sivuston taitossa ei tarvitse käyttää aikaa painikkeen ylimääräiseen tyyliin.

## 5.2 Heikkoudet ja uhat

Bootstrap, kuten responsiivinen web-sivusto yleensä, vaatii enemmän aikaa testaukseen, kuin kiinteälevyinen sivusto. Responsiivisuuden testaukseen ei riitä, että web-sivusto tarkistetaan vain selainikkunaa kaventamalla, sillä eri laitteilla voi esiintyä laitekohtaisia virheitä ulkoasun toistossa tai ominaisuuksien toiminnallisuudessa. Vaikka BS on laajasti testattu jo kehittäjien osalta, jonkinasteinen perustestaus web-sivustolle on silti tehtävä.

Bootstrapilla rakennettu web-sivusto ei myöskään ole suunniteltu korvaamaan perinteistä mobiilisovellusta. Esimerkiksi kuvien lataus (`input type="file"`) ei ole tuettu lainkaan useissa puhelimissa. Tämä tarkoittaa sitä, ettei web-sivuston perusidea voi olla tarjota käyttäjälle mahdollisuutta ladata kuvia web-palveluun mobiilisti, vaan sitä varten tarvitaan erillinen mobiilisovellus. BS on nimenomaan responsiivinen framework web-sivustoille, ei mobiilisovelluksiin tarkoitettu työkalu.



Mikäli ei halua, että oma web-sivusto huutaa katsojalle Bootstrappia, se vaatii poikkeuksetta oman CSS tyylin kirjoittamista. Ei välttämättä ole huono asia että BS:n oletustyyli näkyvät web-sivuston ulkoasussa, jos sivustolla ei haluta sen kummemmin erottua massasta tai luoda jotain persoonallista. BS vaatii myös jonkin verran käsin tehtäviä CSS korjauksia, jotta esimerkiksi sivuston kaikki kuvat saadaan automaattisesti responsiiviseksi.

Bootstrap teki suuria muutoksia koko grid-järjestelmänsä perusteisiin ja määriteluokkien nimeämiseen päivittäessään versiota 2 versioon 3. Yksi merkittävä asia miksi muutoksiin oli, että BS3 versiossa otettiin huomioon paremmin mobile first suunnittelu. Ongelmaksi näiden kahden eri version kohdalla nousi vanhojen web-sivustojen päivitys. Sivustojen päivitys uudempaan BS versioon ei onnistu vain lataamalla palvelimelle uudet CSS ja JavaScript tiedostot, vaan se vaatii kaikkien templaattien läpikäynnin käsin. Näin suurien muutoksien tekeminen frameworkkiin ei ole hyvästä, sillä web-sivustot, joilla ei ole rahaa suurien päivityksien tekoon, tippuvat pois kehityksestä. Ilman päivityksiä sivuston tekniikka vanhenee nopeasti.

Bootstrapin grid-järjestelmä ei toimi Internet Explorer 7 ja sitä vanhemmilla versioilla. Vaikka responsiivisuus on `respond.js` lisäosan avulla tuettu myös vanhemmilla IE selaimilla, IE 7 versiossa grid-järjestelmä hajoaa. Jos web-sivuston vaatimuksena on että sen tulee toimia myös IE 7:lla, tämä aiheuttaa lisätyötä suunnittelijalle, koska grid-järjestelmä pitää korjata käsin.

### **5.3 Yhteenveto: Miksi projektiin kannattaa valita Bootstrap?**

Mikään valmis responsiivinen framework ei ole täydellinen, ei edes Bootstrap. Työkalu, jonka tarkoituksena on tarjota kattavasti erilaisia tekniikoita web-sivustojen rakentamiseen, ei voi millään kohdata jokaista sivustokohtaista vaatimusta. Tämän vuoksi BS:n helppo kustomointi on todella tarpeellinen, jotta oman web-sivuston haasteet voi korjata.

Vuokraovi.com uudisti sivustonsa 7.4.2014 ja ilokseni huomasin, että he käyttävät Bootstrappia. BS:n käyttö on jonkin verran yleistymässä myös Suomessa, joka tarkoittaa että useampi web-suunnittelija on tutustunut sen tekniikoihin. Web-sivujen ongel-

mana on pitkään ollut se, ettei niiden rakenteen toteuttamiseen ole ollut selkeää, yhdenmukaista tapaa. Jos kaikki web-sivustot alkaisivat käyttää BS:ää, tai jotakin muuta suosittua frameworkia, sivujen ylläpito olisi helpompaa, kun päivittäjällä olisi selkeä dokumentaatio sivuston rakenteesta.

Osa web-suunnittelijoista käyttää mieluummin omaan käyttöönsä rakentamaansa frameworkia, joka sisältää responsiivinen grid-järjestelmän sekä joitakin muita oleellisia tekniikoita. Itse luodun koodin dokumentointi ja kehitys on kuitenkin silloin suunnittelijan vastuulla, eikä alati muuttuvassa web-maailmassa mikään ole ikuista. Bootstrapissa suurimman kehitystyön tekee valmis yhteisö sekä käyttäjät, jotka haluavat jakaa itse hyväksi todettua koodia vapaasti muiden käyttöön.

Loppujenlopuksi aika on rahaa. Bootstrap säästää huomattavasti web-sivuston taittoon käytettävää aikaa, jolloin web-sivustoista saadaan asiakkaan kukkarolle sopivampia toteuttaa.

## 6 WEB-SUUNNITTELUN TRENDIT

Tässä luvussa käsitellään millaisia trendejä tulevaisuudessa web-suunnittelussa voidaan nähdä ja miten Bootstrap niihin vastaa. En käsittele trendeissä lainkaan responsiivista tai mobile first suunnittelua, sillä koen että ne ovat tänä päivänä oletusarvoisesti web-suunnittelun lähtökohtia, ei tulevaisuudessa yleistyviä trendejä.

Web on alati kehittyvä media, jossa myös muiden medioiden tapaan on havaittavissa selkeitä trendejä. Uusien tekniikoiden ja suunnittelukonseptien yleistymisen myötä useilla web-sivustoilla on havaittavissa samankaltaisia elementtejä. Käyttäjät kuitenkin loppupeleissä määräävät, ovatko nämä trendit nousevia vai ei, eli kuinka sivuston kävijämäärät reagoivat muutoksiin.

Tässä luvussa olen käyttänyt lähteenä useita eri artikkeleja (Rocheleau; Leigh Turner 2013; Doru 2014; Lake 2014; Taheri 2014), joita olen vertaillut keskenään ja nostanut niistä viisi, omasta mielestäni oleellisinta trendiä.

### 6.1 Responsiiviset frameworkit

Käyttäjät vaativat jo nyt web-sivustoilta yhdenmukaista käyttökokemusta päätelaitteestaan riippumatta. Koska käsin rakennettava responsiivisuus vie usein paljon aikaa ja on kallis toteuttaa, responsiivisten frameworkkien käyttö tulee yleistymään. Frameworkkeilla web-sivustoista saadaan kustannustehokkaita ja laadukkaita pienellä vaivalla, jolloin asiakkaille voidaan myydä responsiivinen sivusto lähes samaan hintaan kuin perinteinen, kiinteälevyinen sivusto.

Bootstrapin käyttö tulevaisuudessa tulee kasvamaan entisestään, sillä se on helppo oppia, monipuolinen käyttää ja se taipuu hyvin erityyppisiin web-projekteihin. BS:n vahvuutena on sen selkeä dokumentaatio ja aktiivinen kehittäjäyhteisö, joka haluaa pitää sen modernina ja toimivana.

Artikkeleissa, joita vertailin (Rocheleau; Leigh Turner 2013; Doru 2014; Lake 2014; Taheri 2014.) ei puhuttu suoranaisesti frameworkkien yleistymisestä, vaan responsiivisista ja mobile first konseptilla suunnitelluista web-sivustoista. Tästä päättelin, että jos

responsiiviset web-sivustot tulevat yleistymään, tulee myös frameworkkien käyttö kasvamaan niiden kustannustehokkuudesta johtuen.

## 6.2 Flat design

Kun vielä muutama vuosi sitten suunnittelijat käyttivät sivuilla paljon erilaisia liukuvärejä tehosteena sekä 3D efektejä mm. painikkeissa, nykyään on alettu suosimaan flat designia eli tasaista suunnittelua. Flat design on minimalistisesta suunnittelusta hieman pidemmälle viety konsepti, mutta samalla idealla sen on tarkoitus pitää web-sivusto yksinkertaisena ja modernina. Flat design keskittyy pääasiassa web-sivuston sisältöön, eikä sen ulkoasuun (content first). Doru (2014) kuvailee osuvasti artikkelissaan että flat design pakottaa web-suunnittelijat saavuttamaan enemmän vähemmällä. Liukuvärien poisjättäminen saattaa nopeuttaa hieman sivuston lataamista, mutta tämä ei ole oleellista. Pääasiassa flat design nopeuttaa sivuston rakentamista, kun elementeille ei tarvitse CSS:llä askarrella erilaisia liukuvärejä, koho- ja heittovarjoefektejä.

Bootstrap ennakoi flat design trendin jo julkaistessaan version 3, sillä uusimmasta versiosta oli jätetty kokonaan pois navigaatiopalkin ja painikkeiden oletusarvoiset liukuvärit, jotka löytyivät vielä versiosta 2.

## 6.3 Typografia

Mobile first suunnittelun vuoksi web-sivustoilla aletaan kiinnittää enemmän huomiota mielenkiintoiseen typografiaan, monimutkaisen grafiikan sijasta. Tekstit ovat nopeampi ladata web-sivustolle kuin kuvat, jolloin lyhempi latausaika parantaa käyttökokemusta. Ilmaisten web-fonttien, kuten Google fonttien käyttö helpottaa typografian toteutusta, sillä niitä ei tarvitse muuttaa erikseen web-fonteiksi.

Bootstrap tarjoaa hyvin kattavan tyylivalikoiman web-sivuston typografian luontiin. Valmiiksi tyylitelty typografia on hillitty ja helposti mukautettava. Määriteluokkien avulla otsikkotasoihin saadaan pienellä vaivalla eri ulottuvuuksia ja näin typografiasta saa mielenkiintoisen.

## 6.4 Single page -suunnittelu

Omaakohtaisesti olen huomannut asiakkaideni vaatimuksissa toistuvan toiveen: web-sivustoista halutaan single page eli yksisivuinen toteutus. Sivustolla saattaa silti olla tarve normaaleille alisivuille, mutta vähintään etusivun pitää olla eri mielenkiintoisesti osioihin jaettu, single page tyyppinen sivu. Perinteinen single page sivusto ei sisällä lainkaan alisivuja, vaan käyttäjää kuljetetaan etusivulla eri osioissa. Osiot ovat rakennettu allekkain ja sivustoa vierittämällä käyttäjä liikkuu osiosta toiseen. Single pagen suosio voi osittain selittyä mobiilikäyttäjien kasvusta. Vierittäminen on mobiilissa helpompaa kuin navigaatiosta linkin klikkaaminen. Borowska (2014) kuvaa ytimekkäästi single page sivuston idean: Enemmän vierittämistä, vähemmän klikkauksia.

Bootstrapin 12-palstaisella grid-järjestelmällä on helppo luoda ja ylläpitää single page tyyppisiä web-sivustoja. Lisäksi kiinnitetty navigaatio soveltuu hyvin single page sivustolle. Kun navigaatiopalkille antaa määriteluokan `.navbar-fixed-top`, se kulkee automaattisesti käyttäjän vierityksen mukana sivustolla.

## 6.5 Videot elävöittävät web-sivustoa

Modernien, HTML5:sta tukevien selaimien yleistyessä suunnittelijoiden tavat elävöittää web-sivustoa ovat kasvaneet. Videoita on ollut jo jonkin aikaa helppo upottaa sivustolle erilaisista web-palveluista, kuten Vimeosta tai Youtubesta. Uusi trendi on se, että suuren taustakuvien tilalla on alettu käyttää videoita. HTML5 video elementillä sekä sekoittamalla sekaan hieman CSS:ää, videosta saa koko sivun levyisen taustaelementin web-sivustolle. Kuva kertoo enemmän kuin tuhat sanaa mutta video kertoo vielä enemmän.

Bootstrap ei tällä hetkellä tarjoa minkäänlaista tukea videoiden käyttämiseen web-sivuston taustakuvana. Mikäli tämä trendi vakiinnuttaa paikkansa, en näe syytä miksi se ei videotukea tulisi jossain vaiheessa tarjoamaan.

## 7 JUHLAPELIT.NET SIVUSTON UUDISTUS

Tässä luvussa käsitellään juhlapelit.net sivuston uudistusta. Aluksi käydään läpi projektin lähtökohdat ja tekniikan valinta web-sivustolle. Sen jälkeen käymme läpi sivuston teknistä toteutusta ja lopuksi tutkimme, miten uudistus on sivustolla onnistunut.

### 7.1 Projektin lähtökohdat

Juhlapelit.net on vuonna 2011 julkaistu Wordpress-julkaisujärjestelmän päälle rakennettu web-sivusto. Julkaisu on siis tapahtunut samoihin aikoihin, kuin responsiivinen suunnittelu alkoi pyöriä suunnittelijoiden huulilla. Juhlapelit.net sivuston tarkoitus on tarjota ilmaisia peli-ideoita ja ajanvietettä aikuisten juhliin. Sivustoa alun perin suunnitellessa se ei ollut responsiivinen, vaan suunnattu yli 960 pikseliä leveille näytöille, eli pääasiassa työpöydille.



KUVA 20. Juhlapelit.net sivuston alkuperäinen ulkoasu

Kävijätietoja seurattessani huomasin, että käynnit mobiililaitteilla ovat alkaneet kasvaa web-sivustollani. Punnitsin olisiko tarve muuttaa se responsiiviseksi. Milloin

käyttäjät todennäköisesti käyvät web-sivustolla? Olin aiemmin seurannut että suurimmat kävijäpiikit olivat Googlen kautta viikonloppuisin, mutta en ottanut huomioon, ovatko käyttäjät silloin kotona tietokoneen ääressä vai eksyvätkö he sivustolle, kun ovat poistuneet kotoa. Kun käyttäjä on liikkeessä, esimerkiksi ystävän luona juhlimassa, on todennäköisestä että hän etsii älypuhelimellaan nopeasti tietoa Internetistä. Vaikka tietokone olisi saatavilla, älypuhelimien käyttö nopeaan tiedonhakuun on yleistynyt. Tämän johtopäätöksen vuoksi sivuston responsiiviseksi muuttaminen tuli ajankohtaiseksi.

Pohdin pitkän aikaa, milloin web-sivuston uudistus olisi paras toteuttaa. Omista suunnitelmistani poiketen Wordpress asetti minulle tiukan aikataulun päivittyessään automaattisesti. Sivusto oli suhteellisen vanha ja rakennettu aikoinaan twentyeleven teeman lapsiteemaksi. Kun WP päivitti itsensä uusimpaan versioon, se ei enää tukenut twentyeleven teemaa ja silloinen juhlapelit.net teema hajosi. En usko että olin muutenkaan tehnyt kyseistä lapsiteemaa kaikkien oppikirjojen mukaisesti, jonka vuoksi sen korjaaminen ei ollut vaihtoehto. Sivusto piti saada takaisin toimintaan ja nopeasti! Tästä syystä juhlapelit.net sivuston uudistus eteni ripeästi.

### **7.1.1 Tekniikan valinta**

Koska juhlapelit.net sivusto oli alun perin julkaistu Wordpressiin, en nähnyt syytä vaihtaa sen julkaisualustaa. Valitsin Bootstrapin frameworkiksi sivustolle kolmesta eri syystä. Ensinnäkin BS oli minulle jo tuttu töiden puolesta, jonka vuoksi sen mahdollisuudet ja haasteet olivat jo tiedossa. Toiseksi BS soveltuu hyvin käytettäväksi Wordpress -sivustoon, sillä WP:ssä on mahdollista lisätä omat CSS ja JavaScript tiedostot, sekä muokata sivujen templaattien rakenne BS:n vaatimaan rakenteeseen. Viimeinen, mutta tärkein valintakriteeri Bootstrapille oli sen laadukas grid-järjestelmä, sekä mobile first lähestymistapa sivuston rakentamiseen. Koska tarkoitukseni oli nimenomaan tarjota mobiilikäyttäjälle mieluisa tapa selata sivustoa, BS3 frameworkkinä oli tähän tehokas työkalu.

Päätin kirjoittaa juhlapelit.net sivuston tyylit Less:illä (CSS-laajennus), koska sen merkkikieli ja rakenne verrattuna normaaliin CSS:ään on mielestäni selkeämpi. Less:issä tyylimäärittelyjä voi kirjoittaa toisiensa sisään, joka poistaa pitkien CSS kohdentamisien käytön tyylin alussa, sekä se mahdollistaa muuttujien (variables) käytön.

**custom.less**

```
.navbar{
  .navbar-brand{
    font-size: 36px;
    text-transform: uppercase;
    padding: 25px 15px;
    color: @grey-fbf;
    &:hover{
      color: @green;
      .green{ color: @white; }
    }
  }
}
```

**custom.css**

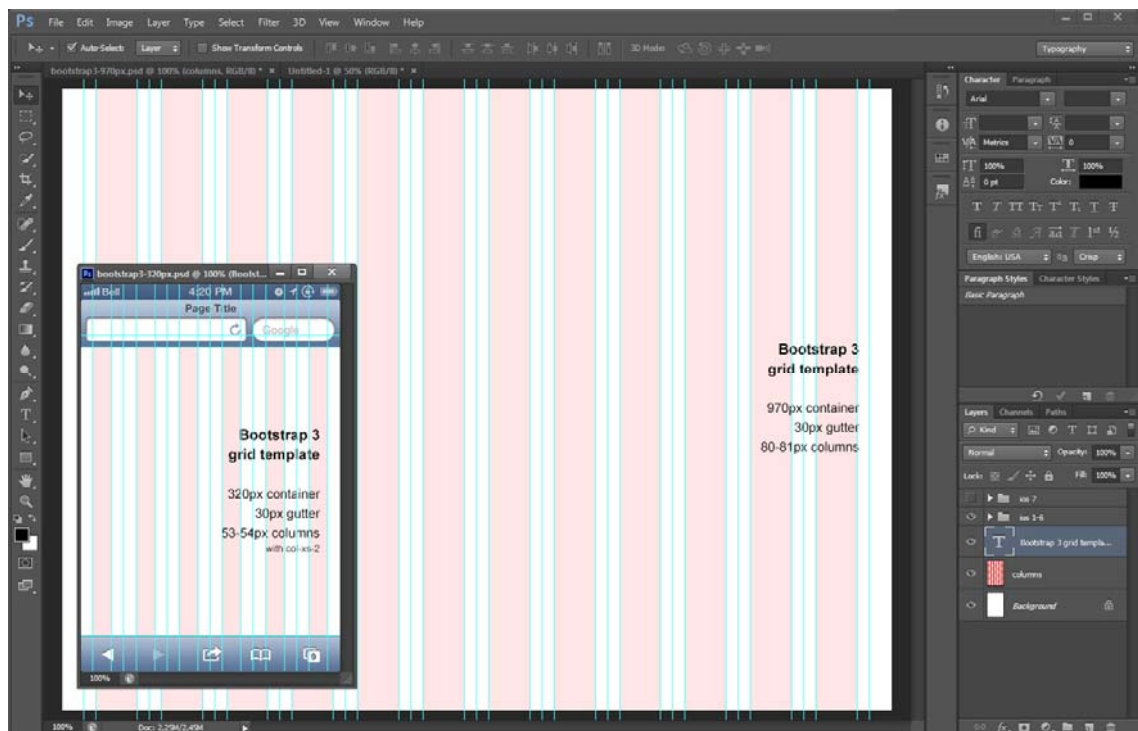
```
.navbar .navbar-brand {
  font-size: 36px;
  text-transform: uppercase;
  padding: 25px 15px;
  color: #fbfbfb;
}
.navbar .navbar-brand:hover {
  color: #49cc21;
}
.navbar .navbar-brand:hover .green {
  color: #ffffff;
}
```

LESS vaatii konvertoinnin CSS muotoon, jonka vuoksi valitsin työkaluksi koodaamiseen ohjelman nimeltään Brackets. Bracketsiin saa asennettua lisäosan, joka konvertoi LESS tiedostot CSS muotoon reaaliajassa. Tyylit kirjoitetaan `custom-style.less` tiedostoon, josta ne päivittyvät automaattisesti `custom-style.css` tiedostoon. Kun uudet tyylit halutaan päivittää web-sivustolle, palvelimelle ladataan `.css` tiedosto.



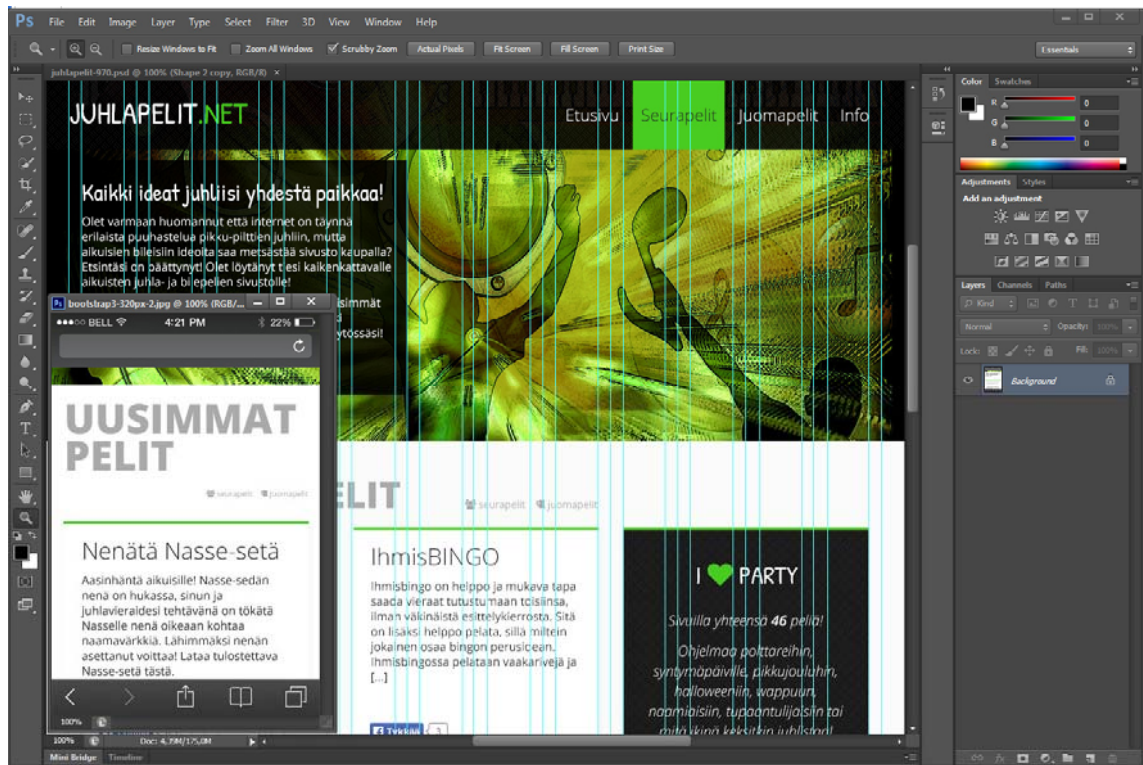
## 7.2 Suunnittelu

Sivuston suunnittelun aloitin asettamalla itselleni päätavoitteen. Sivuston tulee olla mobiiliystävällinen, mutta näyttää hyvältä myös työpöytäkäyttäjille. Latasin Internetistä kaksi BS3 grid-järjestelmään suunniteltua .psd pohjaa. Adobe Photoshop pohjissa oli valmiiksi piirretty 12-palstainen grid-järjestelmä, joissa palstoille on merkitty 15 pikselin levyiset palstatäytteet. Toinen tiedostoista oli tarkoitettu pienille näyttöpinta-aloille, eli 320 pikseliä leveälle näytölle ja toinen 1024 pikseliä leveälle näytölle.



KUVA 21. Valmiit .psd pohjat 320 ja 1024 pikseliä leveille näytöille

Mobile first -konseptia hyödyntämällä suunnittelin ensin, mitä ja miten mobiilikäyttäjälle näytetään sivuston sisältö. Tärkeintä on tarjota käyttäjälle helppo tapa tutustua peleihin ja suuret painikkeet, joita on helppo klikata.



KUVA 22. Juhlapelit.net sivuston suunnittelu Adobe Photoshopissa 320 ja 1024 pikseliä leveille näytöille

Sivuston typografia koostuu kahdesta eri fontista: Patrick Hand ja Open Sans. Olemmat fontit ovat ilmaisia Google fontteja ja ne ovat suunniteltu nimenomaan web-sivustoilla käytettäviksi. Open Sans fontti oli minulle tuttu jo aiemmista web-projekteista, ja pidän sitä hyvin luettavana ja monipuolisena fonttina. Patrick Hand oli minulle uusi fontti, jota en ole ennen käyttänyt. Fontti oli mielestäni leikkisä ja sopi hyvin juhlapelit.net sivuston henkeen.

### 7.3 Toteutus

Tässä osiossa käydään läpi, millaisia vaiheita juhlapelit.net sivuston muuttaminen viimeisempään Bootstrap 3.1.1. versioon vaatii.

#### 7.3.1 Alustus

Aloitin web-sivuston toteutuksen lataamalla Wordpress 3.8.3 version. Päätin käyttää web-sivuston pohjana valmista WP teemaa, josta löytyy perustoiminnallisuudet ja BS3

valmius. Löysin teeman nimeltään WP-Bootstrap, joka oli rakennettu BS 3.0.1 versioon. BS:n versio ei ollut teemassa ajan tasalla, mutta valitsin silti sen juhlapelit.net sivuston isäntäteemaksi. Loin WP-Bootstrap teemasta lapsiteeman, jonka aktivoin käytettäväksi juhlapelit.net sivustolle.

### 7.3.2 Templaattien rakennus

Juhlapelit.net sivustoa varten minun tuli rakentaa seuraavat templaatit: etusivu (`index.php`), sisäsivu (`page.php`), 404 sivu (`404.php`), listausnäkyvä pelien kategorioille (`archive.php`) sekä yksittäisen pelin sivu (`single.php`). Koska teemassa sivuston oikea sivupalkki on vimpainalue, tälle alueelle loin oman `sidebar.php` tiedoston.

Kaikkiin templaatteihin ja sivupalkkiin rakensin Bootstrapin grid-järjestelmän. Web-sivusto sisältöalue on jaettu kolmeen palstaan mobiililaitteiden leveyteen saakka (768 pikseliä), josta pienemmillä näytöillä kaikki palstat muuttuvat 100 prosenttia leveiksi.

Sivustolle asennetussa teemassa oli sivupalkkiin luotu hakulomake, jolla sivuston sisäl- löstä voidaan etsiä tiettyjä hakusanoja. Pidin tästä ideasta niin paljon, että suunnitelmis- tani poiketen jätin hakulomakkeen sivuston sivupalkkiin. Hakutuloksien listaamiseen loin uuden templaatin sivustolle (`search.php`), jonne rakensin myös BS:n grid- järjestelmän.

### 7.3.3 Ulkoasun kustomointi

Kustomoin Bootstrapin sivustolla ladattavat tiedostot juhlapelit.net sivustoa varten. Si- vustolla ei ole käytössä mitään monimutkaisia tekniikoita, joten valitsin ladattavaksi vain tarvitsemäni CSS tiedostot, komponentit sekä jQuery lisäosat. Kustomoimalla BS tiedostot ja käyttämällä tiedostoista pakattuja `.min` versioita web-sivuston latausaikaa voi lyhentää. Latasin kustomoidun `bootstrap.min.css` ja `bootstrap.min.js` tiedostot lapsiteemaan ja loin linkityksen tiedostoihin lapsiteeman `functions.php` tiedostossa.

En käyttänyt juhlapelit.net sivustolla BS:stä löytyviä ikoneita (glyphicons), koska niiden valikoima ei ollut tarpeeksi kattava. Ikoneina käytin ilmaista ikonisettiä nimeltään Font

Awesome, koska kyseisestä ikonisetistä löytyi oluttuopin ikoni, jota web-sivustolla käytän juomapelien yhteydessä. Font Awesome ikonit saa käyttöön yksikertaisesti lataamalla `font-awesome.css` tiedoston sivuston head osiossa.

```
<link href="//netdna.bootstrapcdn.com/font-awesome/4.0.3/css/font-awesome.css" rel="stylesheet">
```

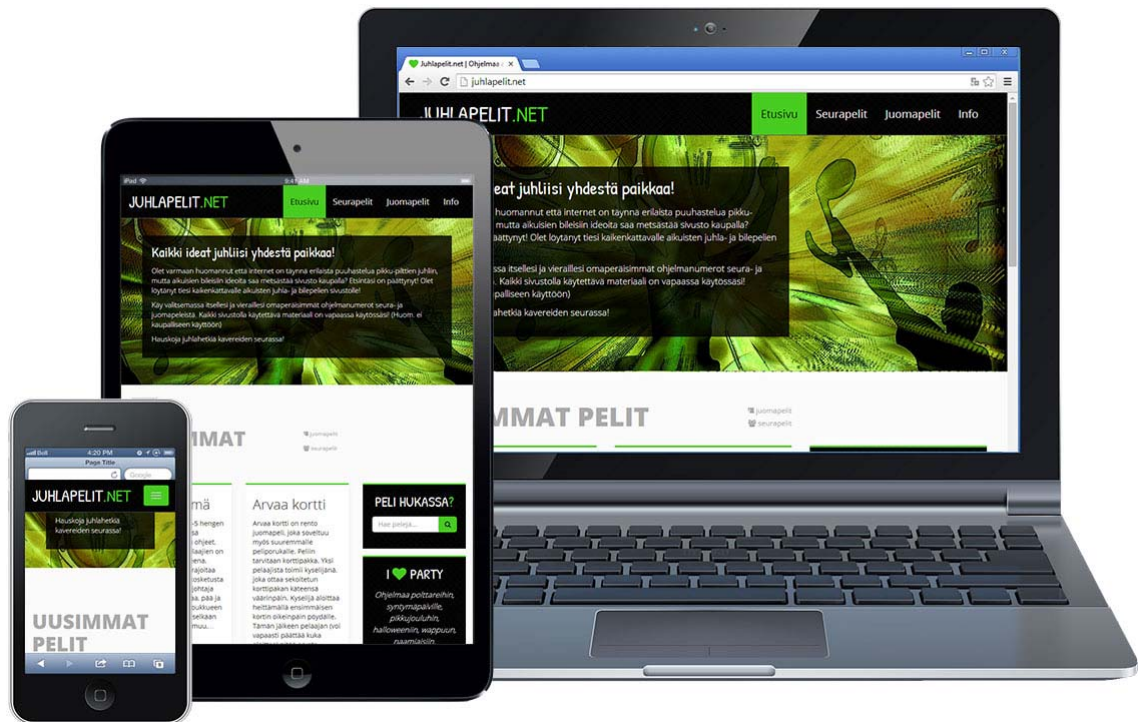
Valitsin juhlapelit.net sivustolle käytettäväksi Google fontit Patrick Hand ja Open Sans. Nämä saa ilmaiseksi käyttöön valitsemalla fontit ja haluamansa leikkaukset Google Fonts sivustolla ja kopioimalla niiden `@import` koodin `custom.less` tiedoston alkuun. Patrick Hand fontista löytyy vain yksi leikkaus (Normaali 400), mutta Open Sans fontissa leikkauksia voi valita useita erilaisia. Käytän projektissa neljää eri leikkausta fontista: kevyt 300, normaali 400, lihavoitu 700 ja ekstra lihavoitu 800.

```
@import url(http://fonts.googleapis.com/css?family=Patrick+Hand|Open+Sans:400,700,800,300);
```

Jotta pääsisin muuttamaan web-sivuston ulkoasun mieleisekseni, tarvitsin `custom.less` tyyli-tiedoston. Kuten aiemmin mainitsin, loin sivuston tyyli-tiedoston Less:llä, koska pidän sen merkkikieltä selkeämpänä normaaliin CSS:ään verrattuna. `custom.less` tiedosto konvertoituu Bracketissä automaattisesti `custom.css` tiedostoksi, joka lisäsin sivustolle ladattavaksi `bootstrap.min.css` tiedoston jälkeen. `custom.less` tiedostossa ylikirjoitin Bootstrapin oletustyylejä ja värejä, jotta sain web-sivuston ulkoasun näyttämään samalta kuin Photoshopissa tehdyssä suunnitelmassani. En käy läpi, mitä kaikki tyylejä ylikirjoitin, sillä `custom.less` tiedosto on lähes 450 riviä pitkä. Tyyli-tiedosto on luettavissa juhlapelit.net sivuston lähdekoodista.

## 7.4 Lopputulos

Juhlapelit.net sivuston uudistus julkaistiin virallisesti aamuyöllä 15.3.2014. Tämän jälkeen olen tehnyt siellä pieniä korjauksia ja muutoksia HTML merkkikielen. Uudistus piti tehdä nopealla aikataululla, jonka vuoksi Bootstrap oli siihen ihanteellinen työkalu.



KUVA 23. Juhlapelit.net eri päätelaitteilla

Olen tyytyväinen siihen, miltä juhlapelit.net sivusto tuli loppujenlopuksi näyttämään. En joutunut tekemään kompromisseja Photoshopissa suunnitellun ulkoasun ja sivuston rakenteen kanssa, koska käytin lähtökohtana nimenomaan Bootstrappia. Koska Bootstrapin tekniikka ja rajoitukset olivat minulle jo tuttuja, pystyin vaivatta hahmottaan miten sivusto tulee näkymään eri näyttötarkkuuksissa.

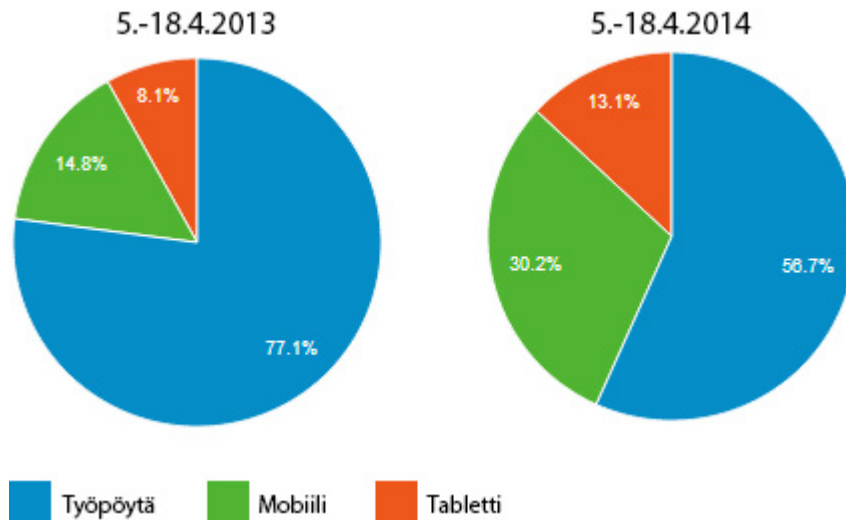
Jatkossa tulen suorittamaan juhlapelit.net sivustolle normaaleja ylläpitotoimia ja pitämään Bootstrapin version ajan tasalla, sekä tietysti luon sivustolle uutta sisältöä.

### 7.4.1 Analytiikka

Kiinnostavaa oli, miten uusi responsiivinen juhlapelit.net sivusto vaikutti sivuston kävijöiden käyttäytymiseen. En odottanut että web-sivustolla olisi tapahtunut mitään räjähtävää kävijämäärän kasvua, sillä uudistusta ei ole mainostettu mitenkään. Sivustolla on ollut jo aiemmin Google Analytics asennettuna, jonka kautta sain noudettua tietoa kävijäkäyttäytymisen vertailuun.

Ensimmäiseksi vertailin, millä laitteilla kävijät ovat vierailleet web-sivustolla vuosi sitten ja uudistuksen jälkeen. Laitetiedoissa kiinnitin huomiota pääasiassa mobiilikäyntien


kasvuun. Sivuston uudistus ei yksinään selitä mobiilikäyntien määrän kasvamista, vaan siihen vaikuttaa myös mobiililaitteiden yleistyminen Internetin selaamiseen. Vertaillaessa laitetietoja kahden viikon ajanjaksolla nyt ja vuosi sitten, mobiilikäyntien määrä on kasvanut 15.4 prosenttiyksikköä, eli käynnit ovat kaksinkertaistuneet vuoden aikana.



KUVA 24. Juhlapelit.net sivuston kävijöiden laitteet kahden viikon vertailuajalla vuonna 2013 ja uuden sivuston julkaisun jälkeen 2014.

Kävijöiden laitetietoja ehkä oleellisempi tieto on se, kuinka kauan kävijät keskimäärin viettävät sivustolla aikaa. Jos sivusto on huono ja epäkäytettävä, kävijät eivät jää sitä pitkiksi ajoiksi selailemaan. Vertailin keskimääräistä käynnin kestoa juhlapelit.net sivustolla vuosi takaperin ja uudistuksen jälkeen. Iloisena yllätyksenä totesin, että keskimääräinen käynnin kesto oli kasvanut 22.67 prosenttia.

Kun käyntien keskimääräisen keston jakaa laitekohtaiseksi, nousua on tapahtunut kaikilla kolmella laitteella (työpöytä, tabletti ja mobiili). Mobiilissa keskimääräinen käynnin kesto kasvoi noin 12 prosenttia. Työpöytäkäyttäjät viettivät sivustolla jopa kuusi ja puoli minuuttia, eli 36 prosenttia kauemmin kuin vuosi sitten.

Device Category		Avg. Session Duration
		22.67%  00:05:45 vs 00:04:41
1.	tablet	
	Apr 5, 2014 - Apr 18, 2014	00:06:34
	Apr 5, 2013 - Apr 18, 2013	00:05:59
2.	desktop	
	Apr 5, 2014 - Apr 18, 2014	00:06:29
	Apr 5, 2013 - Apr 18, 2013	00:04:46
3.	mobile	
	Apr 5, 2014 - Apr 18, 2014	00:04:02
	Apr 5, 2013 - Apr 18, 2013	00:03:36

KUVA 25. Juhlapelit.net sivustolla käynnin keskimääräinen kesto kahden viikon vertailuajalla vuonna 2013 ja uuden sivuston julkaisun jälkeen 2014.

## 8 POHDINTA

Tavoitteeni tätä opinnäytetyötä tehdessäni oli tarjota lukijalle infopaketti, miksi responsiivinen web-suunnittelu on tänä päivänä tärkeää ja mitä voidaan saavuttaa käyttämällä Bootstrap frameworkkia responsiivisen web-suunnittelun työkaluna. Pääasiassa Bootstrapin tarkoituksena on säästää aikaa, verrattuna alusta saakka itse rakennettavaan responsiiviseen web-sivustoon. Aika on rahaa, eikä sitä voi olla koskaan liikaa käytettävissä web-sivuston suunnitteluun ja toteutukseen.

Haaste opinnäytetyötä kirjoittaessani oli ehdottomasti aiheenrajaus. Koska Bootstrapin läpikäynti vaati paljon alustusta mm. responsiivisesta web-suunnittelusta sekä mobile first suunnittelusta, työstä tuli paljon laajempi, kuin mitä alun perin suunnittelin. Lisäksi halusin käydä läpi tulevaisuuden web-suunnittelutrendejä, koska olen itse kiinnostunut Bootstrapin roolista tulevaisuudessa. Olen kuitenkin melko tyytyväinen opinnäytetyöni laajuuteen, sillä kirjoittaessani opin paljon uusia näkökulmia liittyen yleisesti responsiiviseen suunnitteluun ja vinkkejä BS:n käyttöön. Näitä näkökulmia tulen varmasti hyödyntämään töissäni tulevaisuudessa.

En aiemmin käyttänyt juuri lainkaan Bootstrapin sivustolla olevaa kustomointimahdollisuutta, sillä en kokenut että tiedostot olisivat niin suuria, että ne vaikuttaisivat oleellisesti web-sivuston latausaikaan. Kuitenkin mobile first tekniikoihin tutustuessani tajusin, ettei ole järkeä ladata web-sivustoille sellaista tekniikkaa, mitä en sivustolla tarvitse. BS:ään kuin kuitenkin mahdollista jälkikäteen lisätä tarvitsemiaan tyylejä ja komponentteja, joten ei ole millään tavalla perusteltua, että web-sivuston latausaikaa uhraata ”ehkä joskus tarvitsen” ideologialle.

Olin tyytyväinen miten juhlapelit.net sivuston uudistus onnistui. Vaikka aikataulu oli tiukka, Bootstrapin avulla web-sivuston sai uudistettu hyvin lyhyessä ajassa. Juhlapelit.net sivuston kävijöiden käyttäytymisessä näkyi uudistuksen jälkeen selkeä muutos käyntien keskimääräisessä kestossa sekä mobiilikäyntien määrässä, joka oli minulle positiivinen yllätys. Web-sivuston uudistusta ei mainostettu missään, joka vuoksi itse kävijämäärät eivät muuttuneet.

Jatkan Bootstrapin käyttöä tulevaisuudessa ja kannustan mahdollisimman monia muita web-suunnittelijoita tutustumaan kyseiseen frameworkkiin.



## LÄHTEET

Awwwards. 2013. What are Frameworks? 22 Best Responsive CSS Frameworks for Web Design. Päivitetty 20.2.2013. Luettu 14.4.2014. <http://www.awwwards.com/what-are-frameworks-22-best-responsive-css-frameworks-for-web-design.html>

Bootstrap. n.d. Luettu 22.3.2014. <http://getbootstrap.com/>

Cochran, D. 2012. Twitter Bootstrap web development how-to: a hands-on introduction to building websites with Twitter Bootstrap's powerful front-end development framework. UK: Packt Publishing Ltd.

Datta, A. 2012. Be Responsive: A History of Responsive Design. Julkaistu 12.3.2012 Luettu 21.3.2014. <http://shout.setfive.com/2012/03/12/be-responsive-a-history-of-responsive-design>

Doru, B C. 2014. Top 10 web design trends for 2014. Julkaistu 22.1.2014. Luettu 19.4.2014. <http://99designs.com/designer-blog/2014/01/22/top-10-web-design-trends-for-2014/>

Foundation. 2014. Luettu 19.4.2014. <http://foundation.zurb.com/>

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

GitHub, 2014. Luettu 10.3.2014. <https://github.com/twbs/bootstrap>

Gube, J. 2013. 10 Best Responsive HTML5 Frameworks and Tools. Julkaistu 2.6.2013. Luettu 18.4.2014. <http://designinstruct.com/roundups/html5-frameworks/>

Lake, C. 2014. 18 pivotal web design trends for 2014. Julkaistu 13.1.2014. Luettu 19.4.2014. <https://econsultancy.com/blog/64096-18-pivotal-web-design-trends-for-2014>

Leigh Turner, A. 2013. 10 Web design trends you can expect to see in 2014. Julkaistu 29.12.2013. Luettu 19.4.2014. <http://thenextweb.com/dd/2013/12/29/10-web-design-trends-can-expect-see-2014/>

Ma.Cro Mind. 2013. A Brief History of HTML Layouts. Päivitetty 15.6.2013. Luettu 18.4.2014. <http://marcocrosa.wordpress.com/2013/06/15/a-brief-history-of-html-layouts/>

Marcotte, E. 2010. Responsive Web Design. Julkaistu 25.5.2010. Luettu 22.3.2014. <http://alistapart.com/article/responsive-web-design>

Marcotte, E. 2011. Responsive Web Design. New York: A Book Apart.

Open Source Initiative. n.d. The MIT License (MIT). Luettu 22.3.2014. <http://opensource.org/licenses/mit-license.php>

Powers, D. 2011. Introduction to media queries – Part 1: What are media queries? Päivitetty 25.7.2011. Luettu 21.3.2014.  
<http://www.adobe.com/devnet/dreamweaver/articles/introducing-media-queries.html>

Rocheleau, J. Web Design: 20 Hottest Trends To Watch Out For In 2014. Luettu 19.4.2014. <http://www.hongkiat.com/blog/web-design-trends-2014/>

Taheri M. 2014. Web Design Trends for 2014. Julkaistu 14.1.2014. Luettu 19.4.2014.  
<http://webdesignledger.com/trends-2/web-design-trends-for-2014>

Viswanathan, P. 2014. What is a Mobile Device? [ei julkaisu tai päivitysaikaa]. Luettu 30.3.2014. <http://mobiledevices.about.com/od/glossary/g/What-Is-A-Mobile-Device.htm>

W3Counter, Global Web Stats. 2014. Päivitetty 28.2.2014. Luettu 8.3.2014.  
<http://www.w3counter.com/globalstats.php>

W3C. 2014. Media Queries Publication History. Luettu 21.3.2014.  
<http://www.w3.org/standards/history/css3-mediaqueries>

W3C. 2012. Media Queries. Julkaistu 19.6.2012. Luettu 14.4.2014.  
<http://www.w3.org/TR/css3-mediaqueries/>

Wroblewski, L. 2011. Mobile first. New York: A Book Apart.