

Janne Rättyä

Tehtävänsuorittajien ja esikäsittelijöiden käyttö verkkosivukehityksessä

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Mediatekniikan koulutusohjelma

Insinöörityö

12.5.2015

Tekijä Otsikko Sivumäärä Aika	Janne Rättyä Tehtävänsuorittajien ja esikäsittelijöiden käyttö verkkosivukehityksessä 34 sivua + 3 liitettä 12.5.2015
Tutkinto	insinööri (AMK)
Koulutusohjelma	mediatekniikka
Suuntautumisvaihtoehto	graafinen tekniikka
Ohjaaja	yliopettaja Harri Airaksinen
<p>Insinööriyön tarkoituksena oli uudistaa kuntoutusyhdistyksen verkkosivut ajan tasalle. Yhdistyksen tavoitteena on yhdistää kuntoutuksen parissa työskenteleviä ihmisiä, mutta sen verkkosivut olivat vanhentuneet eikä sillä ollut toimivia sosiaalisen median kanavia.</p> <p>Työssä perehdyttiin nykyaikaisen verkkosivun vaatimuksiin, mikä sisältää ulkonäölliset, toiminnalliset ja sisällölliset asiat. Työssä käytiin läpi myös verkkosivukehityksen työnkulkua ja sen kehittämiseksi ja nopeuttamiseksi luotuja JavaScript-apuvälineitä. Näitä apuvälineitä ovat esimerkiksi Grunt, Gulp, Bower, Less ja Sass, ja ne toimivat lokaalisti Node.js JavaScript -suoritusympäristössä.</p> <p>Sivusto toteutettiin Wordpress-sisällönhallintajärjestelmään. Pohjana käytettiin muokattua Roots-teemapohjaa ja Bootstrap-apukirjastoa. Sivuston kehitystyö tehtiin lokaalisti XAMPP-kehitysympäristössä. Kehitystä varten asennettiin myös Node.js-suoritusympäristö ja Grunt-tehtävänsuorittaja ja sen tarvitsemat lisäosamoduulit. Kehitystyössä käytettiin myös Less-esikäsittelijää ja jQuery-apukirjastoa.</p> <p>Lopputuotteena syntyi valmis responsiivinen Wordpress-teema, joka asennettiin asiakkaan palvelimelle. Sisällönhallintajärjestelmään syötettiin asiakkaan luoma sisältö ja tuotiin erillisessä Blogger-palvelussa ollut kuntoutusblogi. Verkkosivun lisäksi yhdistykselle tehtiin uusi logo, suunniteltiin yhtenäinen verkkoilme, luotiin sosiaalisen median kanavat Twitteriin ja Facebookiin sekä uutiskirjepohja MailChimp-palveluun. Asiakkaalle luotiin ohjeistus Wordpressin käyttöön ja annettiin käyttäjätunnukset sosiaalisen median käyttöä varten.</p>	
Avainsanat	Wordpress, verkkosivu, kehitys, suunnittelu, JavaScript, Node, Grunt, Less

Author Title	Janne Rättyä Task runners and preprocessors in web development
Number of Pages Date	34 pages + 3 appendices 12 May 2015
Degree	Bachelor of Engineering
Degree Programme	Media Technology
Specialisation option	Graphic Technology
Instructor	Harri Airaksinen, Principal Lecturer
<p>The purpose of this final year project was to redesign the website of a rehabilitation association. The aim of the association is to connect the people working in the field of rehabilitation, but its website was getting old and it did not have proper social media channels.</p> <p>The thesis analyzes the modern requirements of a web page, which includes aesthetic, functional and content issues. The work looks into the web development workflow and how to improve and speed it up with JavaScript tools. These tools are, for example, Grunt, Gulp, Bower, Less and Sass, and they work locally in the Node.js JavaScript runtime environment.</p> <p>The website was made using the WordPress content management system and a modified skeleton theme named Roots and the Bootstrap library. The development work was carried out locally on the XAMPP development environment. For development purposes the Node.js runtime environment was also installed, as well as the Grunt task runner and the plugins it requires. In the development work also the Less preprocessor and the jQuery library were used.</p> <p>The finished product was a responsive WordPress theme, which was installed on the customer's server. Text content to the content management system was imported from the client's old website, and the content of a blog on rehabilitation was imported from Blogger. In addition to the web page, a new logo and web identity were designed and social media channels were created on Twitter and Facebook as well as the MailChimp newsletter template. Also a WordPress guide was created for the client as well as instructions and credentials for the social media accounts.</p>	
Keywords	Wordpress, web, development, design, JavaScript, Node, Grunt, Less

Sisällys

1	Johdanto	1
2	Nykyaikaisen verkkosivun vaatimukset	2
2.1	Visuaalinen ilme	2
2.2	Toimivuus kaikilla laitteilla	2
2.3	Sisältö ja verkkoläsnäolo	6
3	Verkkosivukehityksen työnkulun apuvälineet ja trendit	9
3.1	Palvelinpuolen JavaScript ja Node.js -suoritusympäristö	10
3.2	Apukirjastot sekä JavaScript- ja CSS-esikäsittelijät	10
3.3	Grunt- ja Gulp-tehtävänsuorittajat	12
3.4	Git-versionhallinta	14
4	Verkkosivun toteutuksen työnkulku	16
4.1	Projektin aloitus	16
4.2	Visuaalinen ilme ja rakennesuunnitelma	19
4.3	Sivuston kehittämisen aloittaminen lokaalisti	22
4.4	Tehtävänsuorittajien ja esikäsittelijöiden käytön aloitus	23
4.5	Julkaisu, testaus ja ohjeistus	29
4.6	Tulokset	29
5	Yhteenveto	32
	Lähteet	33
	Liitteet	
	Liite 1: MailChimp- uutiskirjepohja	
	Liite 2: Valmiin teeman kansiorakenne ja tarvittavat tiedostot	
	Liite 3: Valmis sivusto	

1 Johdanto

Insinööriyön lähtökohtana on uudistaa KUTKE ry:n verkkosivu vastaamaan nykypäivän tarpeita sekä teknisesti että visuaalisesti. Samalla halutaan luoda verkkosivun ilmeen mukaisesti yhdistykselle sosiaalisen median kanavat ja uutiskirje, joiden avulla se voisi tiedottaa ajankohtaisista asioista paremmin ja saavuttaa useampia käyttäjiä.

Kutke on perustettu jo vuonna 1985 nimellä Suomen lääkinnällisen kuntoutuksen yhdistys, mutta nykyisellä nimellään se on toiminut vuodesta 2003. Yhdistyksen toiminnan tavoitteena on edistää vammaisten, vajaakuntoisten ja ikääntyvien henkilöiden kuntoutumista palvelujärjestelmiä kehittämällä. Sen toimintakeinoina on pääasiallisesti lisätä yhteistyötä ja vuorovaikutusta julkaisujen, esitysten, aloitteiden ja koulutusten kautta.

Verkkosivu toteutetaan Wordpress-sisällönhallintajärjestelmään sen edistyneen blogiominaisuuden vuoksi. Itselläni on myös aikaisempaa kokemusta sen käytöstä, joten valinta oli helppo. Verkkosivuun yhdistetään sisältö yhdistyksellä aiemmin käytössä olleesta Blogger-blogista.

Tavoitteenani on luoda verkkosivusta selkeä ja helppolukuinen sen runsaan tekstisisällön takia, mutta kuitenkin visuaalisesti hyvän näköinen ja moderni.

Verkkosivun oheen tarvitaan uutiskirjepohja ja jäseneksi rekisteröitymis- ja uutiskirjeen tilauslomake. Tämä tehdään Mailchimp-palvelun avulla, jonka olen havainnut hyväksi aikaisemmissa projekteissa. Sinne voi lomakkeiden avulla kerätä jäsenrekisteriä tai tilaajia ja lähettää heille valikoidusti kampanjakirjeitä muokattavien pohjien avulla.

Käsittelen raportissa myös nykypäivän verkkosivuprojektin toteuttamisen työnkulkua ja siihen liittyviä apuvälineitä ja koodikirjastoja. Tarkastelen erityisesti Node.js-palvelinympäristössä toimivia JavaScript-tehtävänsuorittajia ja -esikäsittelijöitä.

2 Nykyaikaisen verkkosivun vaatimukset

2.1 Visuaalinen ilme

Kun lähdetään miettimään, mitkä asiat tekevät jostain verkkosivusta nykyaikaisen, tulee yleensä ensimmäisenä mieleen, että se näyttää uudelta. Ulkoasun merkitystä ei voi väheksyä, sillä se on toiminnallisuuden lisäksi se tekijä, jonka ansiosta kävijä kiinnostuu jäämään sivustolle ja joka joko edesauttaa tai hankaloittaa sivuston sisällön hahmottamista [Miller 2011: 89].

Verkkosivun tyylin peruselementteihin sisältyvät väri, tekstuuri, typografia ja kuvakielen käyttö. Yleisesti ottaen tyyliin vaikuttavat ajan trendit ja saatavilla olevat teknologiat, joilla toteuttaa suunniteltava tuote. Näin ollen verkkosivusuunnittelussakin tietokone-tekniikan, selainohjelmiston ja merkkikielen kehitys vaikuttaa suunnittelutyylisiin ja trendeihin. [Miller 2011: 89.]

Nyt suunniteltu verkkosivu saattaa näyttää vanhanaikaiselta jo parin vuoden päästä. Siksi jos haluaa verkkosivusta mahdollisen pitkäikäisen, kannattaa suunnittelussa käyttää hyväksi havaittuja graafisen suunnittelun periaatteita eikä mennä viimeisimpien trendien perässä.

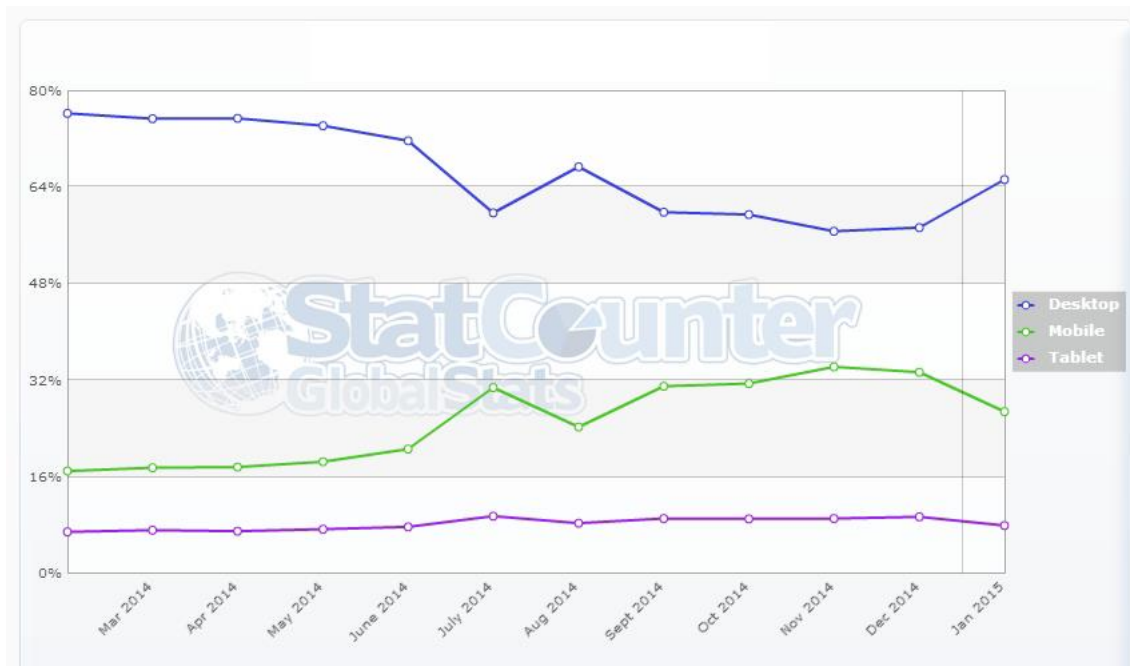
Typografia on osa visuaalista ilmettä ja myös tärkeä osa verkkosivusuunnittelua. Se välittää viestin lukijalle, ja sen muokkaaminen voi joko vahvistaa tai heikentää viestiä. Viestin perille menoon vaikuttaa tekstin luettavuus, johon vaikuttaa kaksi asiaa: kuinka hyvin sen muoto ilmaisee sisällön merkitystä ja kuinka helposti lukija saa selvää luke-
mastaan tekstistä. [Miller 2011: 119.]

Käytännössä ilmaisullisella typografialla saadaan lukijan huomio esimerkiksi verkkosivuston etusivuilla tai otsikoissa, kun taas selkeä typografia edesauttaa luettavuutta teksteissä.

2.2 Toimivuus kaikilla laitteilla

Kuvasta 1 nähdään, että Suomessa lähes 30 % kaikkien verkkosivujen kävijöistä tulee sivuille mobiililaitteella. Tablettien käyttäjiä on lähes 10 %. Tämä tarkoittaa, että yhä

enenevässä määrin tulee ottaa huomioon muutkin kuin perinteisillä tietokoneella nettiä selailevat. [StatCounter Global Stats 2015.]



Kuva 1. Verkkosivuille tulijoiden käyttämät laitteet Suomessa [StatCounter Global Stats 2015].

Tämän vuoksi Kutken verkkosivu päätettiin myös tehdä toimivaksi matkapuhelimille ja tableteille. Tämä toteutettiin tekemällä *responsiivinen* eli mukautuva sivusto, mikä tarkoittaa sitä, ettei tehdä erillistä mobiilisivua, vaan mukautetaan sivu päätelaitteeseen sopivaksi.

Responsiivisuus toteutetaan CSS:n mediakyselyn (engl. media query) avulla, joka esiteltiin CSS3:n myötä. Media queryiden avulla tarkastellaan päätelaitteen ominaisuuksia ja tehdään tarvittavat CSS-muutokset toteutetun kyselyn puitteissa. Yleensä kyselyllä haetaan näyttöresoluutiota, joka vaikuttaa eniten verkkosivun skaalaamiseen. Media queryillä voi myös hakea seuraavia asioita:

- selainikkunan korkeus ja leveys
- laitteen näyttöruudun korkeus ja leveys
- näytön suunta eli onko näyttö pysty- vai vaakasuuntainen (esimerkiksi tableteissa ja matkapuhelimeissa)

- näytön resoluutio ja pikselitiheys [CSS3 @media Rule 2015]

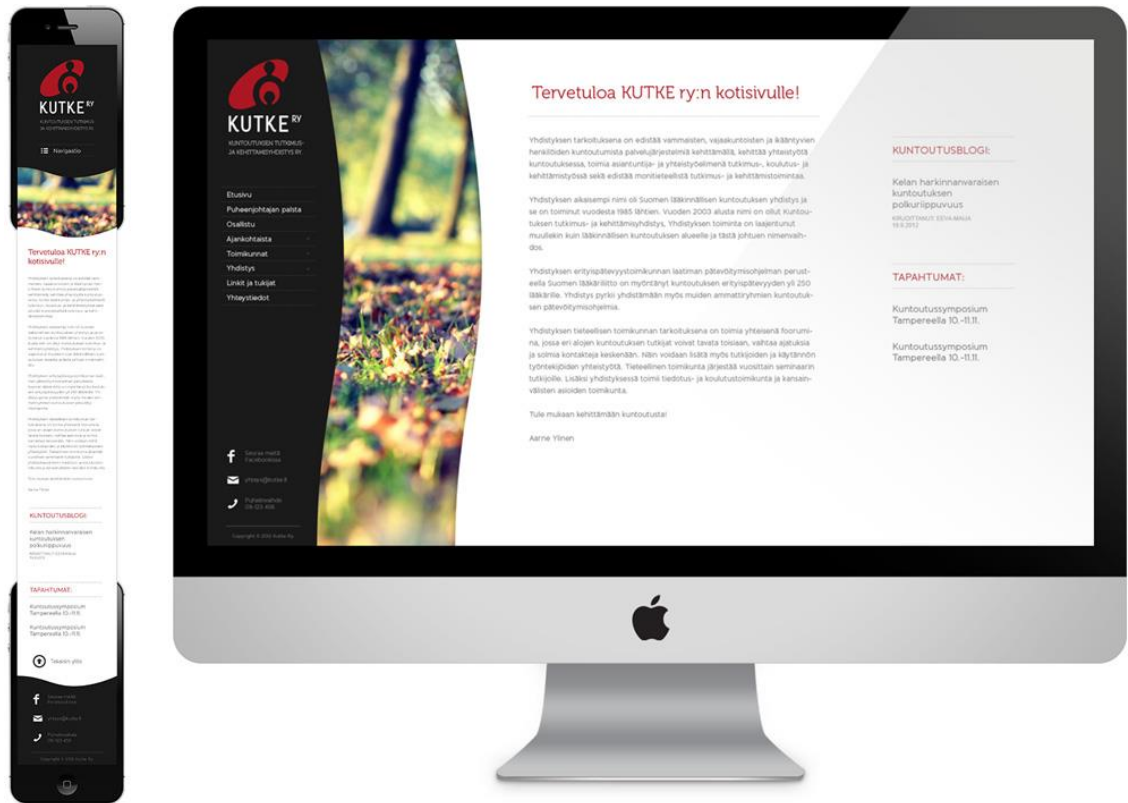
Koodiesimerkistä 1 näkyy kyselyn yksinkertaisin muoto. Kyselyn sisällä olevat CSS-tyylit tulevat voimaan, kun selainikkunan leveys on yli 720 pikseliä.

```
@media (min-width: 720px) {  
  h1 {  
    font-size: 20px;  
  }  
}
```

Koodiesimerkki 1. Media query CSS-tyylitiedoston sisällä.

Projektin tein toteuttaen *mobile first* -suunnitteluperiaatetta, jolloin sivusto rakennetaan ensin sopivaksi mobiililaitteelle ja lähdetään siitä muokkaamaan leveämmille näytöille sopivaksi. Tällä saavutetaan se etu, että vältetään sisällön karsiminen mobiilikäyttäjää varten, mikä usein on tapana [Marcotte 2011: 111]. Sen avulla vältetään myös ylimääräisen CSS:n kirjoittamiselta, kun esimerkiksi tyylitiedostossa aikaisemmin määritettyjä objektien tyylejä ei tarvitse erikseen kumota mobiilileveyttä varten.

Tekemässäni projektissa responsiivisuus näkyy eniten esimerkiksi navigaation toiminnassa ja siinä, kuinka se tulee esiin ja muuttuu ylävalikosta sivuvalikoksi selainikkunan levetessä. Kuvasta 2 nähdään, miltä tämä näyttää ja miten myös sisältöalueen leveys muuttuu näyttöruudun mukaan.



Kuva 2. Verkkosivun responsiivisuus ja mukautuminen mobiilivasta kokoleveään työpöytänäyttöön.

Lukulaiteystävällisyys

Yksi verkkosivun keskeisistä tavoitteista on saavutettavuus (accessibility) [Korpela 2005: 16]. Sivuston saavutettavuus koetellaan viimeistään siinä vaiheessa, kun otetaan kaikki CSS- ja JavaScript-tiedostot pois käytöstä. Lukulaitteet tekevät juuri tämän, ja jäljelle jää puhdas HTML-merkkikieli. Tällöin on hyvä, että verkkosivun semantiikka on kunnossa sekä otsikot, navigaatiot ja kuvat oikeilla paikoillaan. [Coyer 2007.]

Lukulaiteystävällisyyttä voi kehittää ARIA-rooleilla (Accessible Rich Internet Applications). Ne ovat sarja attribuutteja, joita voi lisätä HTML-elementteihin selventämään merkitystä lukulaitteille. Esimerkiksi `<ul role="navigation">...` kertoo, että luettelo on tarkoitettu navigaatioksi. [Kliehm 2007.]

Vaikka ARIA-roolit esiteltiin jo vuonna 2007 ja osan niistä voi nykyään korvata HTML5:n myötä tulleilla semanttisilla elementeillä, ne ovat vieläkin ajankohtaisia [Max 2014].

Projektissa käytetty Roots-aloitusteema sisältää valmiiksi seuraavia Bootstrap-apukirjaston mukana tulleita ARIA-rooleja: banner, navigation, document, main, complementary, contentinfo.

2.3 Sisältö ja verkkoläsnaolo

Sisällön visuaalisuus ja verkkosivun toimivuus eri laitteilla ei kuitenkaan riitä, vaan olennaisia asioita verkkosivussa ovat myös sisältö ja sen tarkoitus sekä siihen käsiksi pääseminen. Dave Lawrencen ja Soheyla Tavakolin mukaan tasapainoisella verkkosivusuunnittelulla on kolme peruspilaria: **käytettävyys** (usability), **tarkoitus** (purpose) ja **esteettisyys** (aesthetics) [Lawrence & Tavakol 2007: 10]. He kertovat, että kunkin näistä tarkka luonne ja laajuus riippuvat kyseessä olevan verkkosivun lajityypistä ja pyrkimyksistä ja selviävät analysoimalla sivun vaatimuksia [Lawrence & Tavakol 2007: 10].

Projektityössä oli paljon tekstisisältöä: blogikirjoituksia ja yhdistysinformaatiota. Yhdistyssivun tarkoituksena on pääosin palvella yhdistyksen jäseniä, jäseneksi harkitsevia tai suurta yleisöä, joka on kiinnostunut yhdistyksessä sen oltua esillä julkisuudessa [Korpela 2005: 142]. Työssä haluttiin esitellä yhdistyksen toimintaa ja alan kuulumisia sekä samalla tehdä kävijälle helpoksi hakea jäsenyyttä.

Sisällön löytämiseen ja omaksumiseen vaikuttaa sivuston käytettävyys. Käytettävyys on termi, joka viittaa siihen, kuinka helppo käyttäjän on oppia, toimia tai saada tarvitsemansa verkkosivun tai ohjelman käyttöliittymän avulla [Miller 2011: 172]. Käytettävyyteen vaikuttavat myös esimerkiksi edellä läpikäytyt asiat toimivuudesta kaikilla laitteilla.

Steve Krugin mukaan tärkein asia käytettävyysuunnittelussa on, ettei laita sivuilla kävijää ajattelemaan turhaan tai tekemään ylimääräisiä asioita. Don't make me think! -virkkeen avulla käytettävyyteen voi kiinnittää huomiota pelkän maalaisjärjen avulla. Mitä vähemmän kävijä joutuu selaamaan sivua löytääkseen haluamansa asian, sitä parempi. [Krug 2006: 11.]

Blogi, hakukoneoptimointi, semantiikka, verkkoläsnaölo

Sisällöstä päästään jouhevasti verkkoläsnaöloon. Ei riitä, että yhdistyksellä on verkkosivu, vaan sen pitää myös näkyä hakukoneissa ja sosiaalisessa mediassa sekä pysyä ajan tasalla ja päivittyä.

Hakukonenäkyvyyteen vaikuttaa esimerkiksi se, kuinka paljon sivuilla on hakusanoihin liittyvää sisältöä, kuinka hyvin hakukone löytää sisällön (semantiikka) ja kuinka paljon muilla sivuilla on linkkejä sivuun. Hakukoneet näkevät verkkosivut pelkkänä tekstinä, joten mitä tekstiystävällisempi sivu on, sitä parempi. Olisi hyvä miettiä tärkeimpiä avainsanoja ja sisällyttää niitä asiatekstiin, otsikoihin, linkkeihin, sivuotsikkoon ja meta-kuvaukseen. Otsikoista voi myös tehdä kuvaavia, jotta ne kertovat sisällöstä mahdollisimman paljon. [Korpela 2005: 146.]

Sosiaalinen media ja sen integrointi

Nykyaikaisissa verkkosivuissa tulee ottaa huomioon, että sivuja halutaan jakaa sosiaalisen median palveluissa, kuten Facebookissa (<https://www.facebook.com/>) tai Twitterissä (<https://twitter.com/>). Näillä molemmilla palveluilla on olemassa oma API eli ohjelmointirajapinta ja metatietoasetukset jakamista varten.

Jos halutaan vaikuttaa siihen, miltä jaettava sivu näyttää julkaisussa ja mitä siinä lukee, on verkkosivuille asetettava palvelua tukevia metatageja sivun head-osioon. Facebook käyttää Open Graph -protokollaa, jonka meta-tagien ominaisuudet ovat muotoa og:title.

Esimerkiksi koodiesimerkissä 2 og:image-tagin kertoo Facebookille, mitä kuvaa halutaan käyttää linkin yhteydessä, kun se jaetaan Facebook-palvelussa.

```
<meta property="og:title" content="Kutke Ry | Kuntoutuksen tut-  
kimus- ja kehittämissyhdistys" />  
<meta property="og:type" content="non_profit" />  
<meta property="og:url" content="http://www.kutke.fi" />  
<meta property="og:image" content="/kutke/wp-  
content/themes/kutke-roots/assets/img/fb-img.jpg" />  
<meta property="og:site_name" content="Kutke Ry" />  
<meta property="og:description" content="Kuntoutuksen tutkimus-  
ja kehittämissyhdistys KUTKE ry on toiminut kuntoutuksen kentällä  
toimivien yhteisenä foorumina vuodesta 2003 alkaen."/>
```

Koodiesimerkki 2. Facebookin käyttämät Open Graph -meta-tagit sivun head-osiossa.

Projektityössä määritettiin Facebookille ja Twitterille metakuvaukset ja kuvat jakamista varten. Sivuille tehtiin myös Facebook-komentointiosio, jossa lukijat voivat kommentoida artikkeleita Facebook-tunnuksillaan. Sen tarkoituksena oli luoda sivuille interaktiomahdollisuutta ja pienentää käyttäjien kynnystä antaa palautetta ja osallistua keskusteluun.

Hakutuloksia ja sosiaalisen median näkyvyyttä voi myös parantaa maksullisilla palveluilla. Google tarjoaa Adwords -palvelua ja mainostilaa, mikä tuo lisänäkyvyyttä määriteltujen hakusanojen avulla. Facebook ja Twitter tarjoavat kohdistettua mainontaa, jolla yrityksen julkaisu tai mainos näkyy kohderyhmän uutisvirrassa. Niiden avulla voidaan saada kohtuullisen helposti ja halvalla lisänäkyvyyttä sivustolle, mutta sitä ei kuitenkaan koettu tarpeelliseksi tässä projektissa.

3 Verkkosivukehityksen työnkulun apuvälineet ja trendit

Nykyaikaisen verkkosivun rakenne on koostunut jo pitkään kolmesta osasta: HTML on sivunkuvauskieli, joka määrittää sivun rungon ja rakenteen, CSS-tyylit antavat rungolle ulkoasun ja JavaScript tuo toiminnallisuutta, sujuvuutta ja erinäköisiä tehosteita ja animaatioita.

Vaikka nämä tekniikat ovat olleet olemassa jo pitkään, ne ovat kukin alallaan kehittyneet toimivammiksi ja tuoneet uusia ominaisuuksia kehittäjien käytettäväksi. HTML päivittyi virallisesti uusimpaan HTML5-versioon lokakuussa 2014, vaikkakin selaimet ovat jo tukeneet sitä pitkään [W3C Recommendation HTML5 2014]. Se toi uusia multimediaominaisuuksia aikaisempaan standardiin, tärkeimpinä ehkä video- ja canvas-elementit, selain- ja sovellusvälimuisti sekä uudet lomakepiirteet ja mikromuodot [Korpela 2011: 13].

CSS:n uusin standardi on CSS3, joka jakoi aikaisemman version pienempiin moduuleihin, joita kehitetään erillään toisistaan. Tämä takaa taaksepäin yhteensopivuuden uusia ominaisuuksia kehitettäessä. CSS3:n tärkeimpiä ominaisuuksia ovat @font-face-fonttien käyttö sekä animaatiot ja transitiot, jotka osaltaan ovat vähentäneet JavaScriptin käytön tarvetta verkkosivuilla [W3C – CSS3 Introduction 2015].

JavaScript on vuonna 1995 kehitetty ohjelmointikieli, jonka avulla verkkosivuihin voidaan luoda dynaamisuutta ja vuorovaikutteisuutta. Netscape kehitti sen alun perin Navigator 2 -selaintansa varten. Kieltä kutsuttiin alussa nimellä LiveScript, mutta se vaihdettiin myöhemmin JavaScriptiksi markkinointiseikkojen vuoksi. [Peltomäki 2004: 2.]

JavaScript on tulkattava kieli, eli sitä voi suorittaa missä tahansa JavaScript-tulkin sisältävässä ohjelmassa, kuten esimerkiksi Internet-selaimessa. Sitä voi sisällyttää HTML-tiedostoihin ja muokata tavallisessa tekstieditorissa, mikä tekee siitä helpon ja nopean kirjoittaa. [Peltomäki 2004: 9.]

Nykyään JavaScriptin ydinkieli on standardoitu ECMA TC9 -komitean toimesta, ja se on nimetty ECMAScriptiksi. Sen uusin versio on ECMAScript 5.1, ja siihen pohjautuvan JavaScriptin uusin versio on 1.8.5. JavaScript on tuettu jokaisen selaimen nykyisessä versiossa. [JavaScript technologies overview 2015.]

3.1 Palvelinpuolen JavaScript ja Node.js -suoritusympäristö

Vuonna 2009 Ryan Dahl esitteli teknologian nimeltään Node.js (<https://nodejs.org/>). Se perustui Googlen Chrome-selaimelle kehittämään V8-JavaScript-tulkkiin, joka sai selaimen toimimaan todella nopeasti. Nodea ei tosin ollut suunniteltu toimimaan selaimessa, kuten JavaScriptin kanssa oli aikaisemmin totuttu, vaan se oli suunniteltu suorittamaan JavaScriptiä palvelimella. [Rauch 2012: 1.]

Node.js on asynkroninen JavaScript-suoritusympäristö, joka on suunniteltu skaalautuvia verkkosovelluksia varten. Sen voi asentaa palvelimelle tai paikallisesti Windows-, Mac- tai Linux-koneille komentotulkiksi lataamalla sen osoitteesta <http://nodejs.org>. Se toimii eräänlaisena kevyenä verkkopalvelimena, joka ei kuluta liikaa resursseja. [Nodejs.org 2015.]

Wordpress-verkkosivukehityksessä Node.js:ä voi käyttää apuna asentamalla sen lokaalisti, jolloin sen kautta voi käyttää apuohjelmia ja moduuleita, kuten Grunt-tehtävänsuorittajaa (<http://gruntjs.com/>) ja Bower-pakkauksenhallintaa (<http://bower.io/>).

3.2 Apukirjastot sekä JavaScript- ja CSS-esikäsittelijät

Nykyaikaisesta sivunkehityksestä puhuttaessa on vaikea olla puhumatta apukirjastoista, pakkauksenhallintajärjestelmistä, tehtävänsuorittajista ja esikäsittelijöistä.

JQuery (<http://jquery.com/>) on JavaScript-kirjasto, jonka tarkoituksena on helpottaa DOM:n muokkausta, tehtävähallintaa, animointia ja AJAX:n käyttöä. Ehkä tärkeimpänä se lisää mahdollisuuden kohdentaa JavaScript-komentoja DOM:n elementteihin käyttäen CSS-selektoreita. [Purewal 2014: 103.]

CSS-apukirjastojen ja kehysten avulla HTML-elementeille voi lisätä valmiita tyylimääriä luokkaselektorien avulla. Tämä saattaa nopeuttaa verkkosivun toteutusta varsinkin prototyypivaiheessa. Suosittuja apukirjastoja ovat esimerkiksi Bootstrap, Foundation, Blueprint, Skeleton, Compass ja Profound Grid. Tässä projektissa käytin apuna Bootstrap-kirjastoa (<http://getbootstrap.com/>).

Monet tällaiset apukirjastot on voitu toteuttaa myös jotain esikäsittelijää varten. Esikäsittelijä (engl. preprocessor) on ohjelma, joka ottaa vastaan tietyn tyyppistä dataa ja kääntää sen toisentyyppiseksi [Howe 2014]. HTML:llä, CSS:llä ja JavaScriptilla voi kaikilla olla esikäsittelijöitä. Niiden tarkoitus on tehostaa työkulkua ja koodin kirjoittamista esimerkiksi poistamalla ylimääräisten merkkien tarvetta (lyhenteet) tai vähentämällä toistoa muuttujien ja *mixinien* avulla.

HTML-esikäsittelijöitä ovat esimerkiksi HAML (<http://haml.info/>), Slim (<http://slim-lang.com/>) ja Jade (<http://jade-lang.com/>). JavaScript-esikäsittelijöitä ovat CoffeeScript (<http://coffeescript.org/>) ja LiveScript (<http://livescript.net/>). Suosituimpia CSS-esikäsittelijöitä ovat Sass (<http://sass-lang.com/>), Stylus (<http://learnboost.github.io/stylus/>) ja tässäkin projektissa käytetty Less (<http://lesscss.org/>).

Koodiesimerkistä 3 nähdään muuttujien toiminta Less-kielessä. Muuttujiin voi tallentaa esimerkiksi värikoodeja, jotka toistuvat useasti eri tyylimäärytyksissä. Esimerkissä nähdään myös, että värikoodiin voi lisätä toista väriä, jonka Less yhdistää uudeksi värikoodiksi.

```
// Esimerkki Less-kielen muuttujista:

@nice-blue: #5B83AD;
@light-blue: @nice-blue + #111;

#header {
  color: @light-blue;
}

// Tämän Less-ohjelma muuttaa CSS-tyyliksi seuraavasti:

#header {
  color: #6c94be;
}
```

Koodiesimerkki 3. Less-esikäsittelijän muuttujien toiminta.

3.3 Grunt- ja Gulp-tehtäväsuoittajat

JavaScript-tehtäväsuoittajat ovat nykyaikaisen verkkosivukehityksen apuvälineitä, jotka suorittavat käyttäjän määrittelemiä automatisoituja toimintoja kehityksen nopeuttamiseksi. Niitä voivat olla esimerkiksi JavaScript-koodin oikeinkirjoituksen tarkistus (JSHint), CSS- ja HTML-tiedostojen minimointi (Uglify) tai selainkohtaisten CSS-prefixien lisääminen tyylitiedostoihin (Autoprefixer). Tehtäväsuoittajia käytetään vain lokaalisti sivun luomisen aikana, eivätkä ne ole enää toiminnassa, kun valmis sivu siirretään palvelimelle.

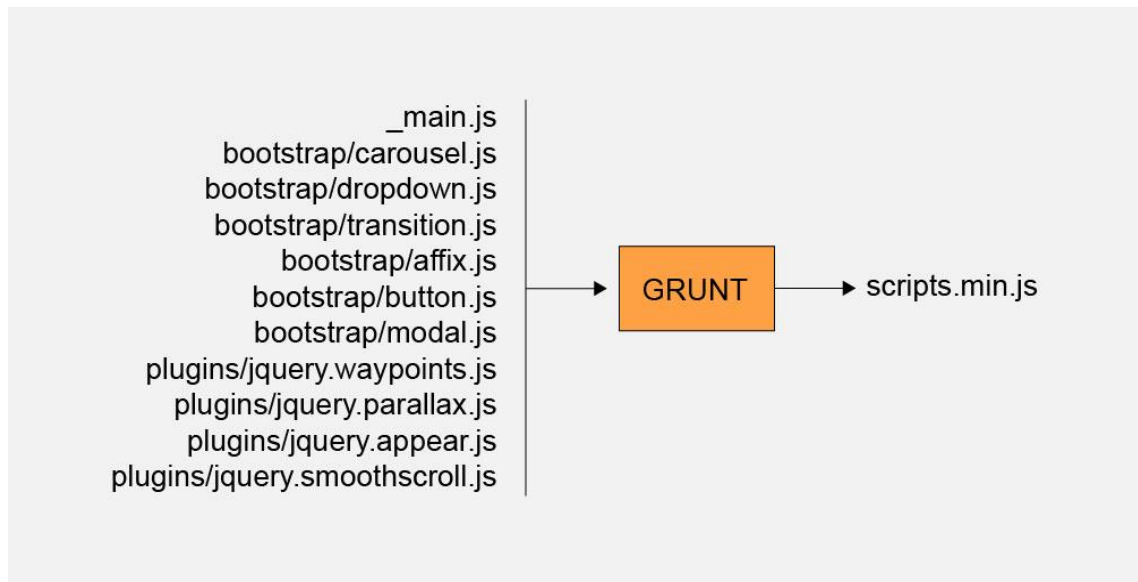
Tätä opinnäytetyötä kirjoitettaessa on olemassa kaksi isoa JavaScript-tehtävän suorittajaa, Grunt (v0.4.5) ja Gulp (v3.8.11, <http://gulpjs.com/>). Grunt on näistä vanhempi ja yleisempi, mutta Gulp on noussut hyvänä haastajana rinnalle yksinkertaisemmän asetustiedoston voimin.

Gruntin ekosysteemi on iso, ja siihen on saatavilla satoja lisäosia. Todennäköisesti tarvittavan lisäosan on joku jo luonut, mutta jos ei ole, sellaisia pystyy luomaan helposti myös itse [Grunt: The JavaScript task runner 2015]. Lisäosia voi selata osoitteessa <http://gruntjs.com/plugins>, ja niiden asennus suoritetaan komentoriviltä, niin kuin Gruntin käyttö muutenkin. Taulukossa 1 on lueteltu muutamia suosittuja Grunt-lisäosia.

Taulukko 1. Suosittuja Grunt-lisäosia ja niiden kuvaus.

Lisäosan nimi	Kuvaus
contrib-watch	Suorittaa ennalta määritettyjä tehtäviä aina, kun kuvioita lisätään, muutetaan tai poistetaan.
contrib-clean	Poistaa tiedostoja ja kansioita.
contrib-uglify	Minimoi tiedostoja käyttäen Uglify.js-ohjelmaa.
contrib-jshint	Tarkistaa JavaScript-tiedostoja virheiden varalta käyttäen JSHint-ohjelmaa.
contrib-less	Kokoaa Less-tiedostot CSS-tiedostoksi.
autoprefixer	Jäsentää CSS-tiedoston ja lisää sen selainkohtaiset etuliitteet tarpeen mukaan.
contrib-imagemin	Pienentää kuvatiedostoja.
contrib-csslint	Tarkistaa CSS-tiedostoja virheiden varalta.

Kuva 3 havainnollistaa Gruntin Uglify-lisäosan toimintaa. Lisäosa hakee tehtävälle määritetyt tiedostot sekä yhdistää ja pienentää niiden sisällöt yhdeksi tiedostoksi.



Kuva 3. Gruntin Uglify-lisäosan toiminta: Grunt kerää määritetyissä kansioissa olevat JavaScript-tiedostot ja yhdistää ja pakkaa ne yhdeksi tiedostoksi.

Gulpia pidetään helpompana käyttää ja kevyempänä asentaa. Se käyttää Noden streaming-ominaisuutta eli komentojen putkitusta (pipes), mikä tarkoittaa, että väliaikastiedostoja ei tarvitse tallentaa levyille, vaan ne voi ohjata putken avulla suoraan toiselle komennolle.

Koodiesimerkissä 4 rekisteröidään Gulp ja kaksi Gulp-lisäosaa (gulp-stylus ja gulp-autoprefixer) ja suoritetaan default-niminen tehtävä. Tehtävä hakee stylus-tiedostot, lisää niihin selainkohtaiset etuliitteet ja kirjoittaa tiedostot uudelleen projektin alakansioon.

```

var gulp = require('gulp'),
    stylus = require('gulp-stylus'),
    autoprefixer = require('gulp-autoprefixer');

gulp.task('default', function() {
  return gulp.src('src/styles/*.styl')
    .pipe(stylus())
    .pipe(autoprefixer())
    .pipe(gulp.dest('public/styles'));
});

```

Koodiesimerkki 4. Gulpfile.js-tiedoston sisältö

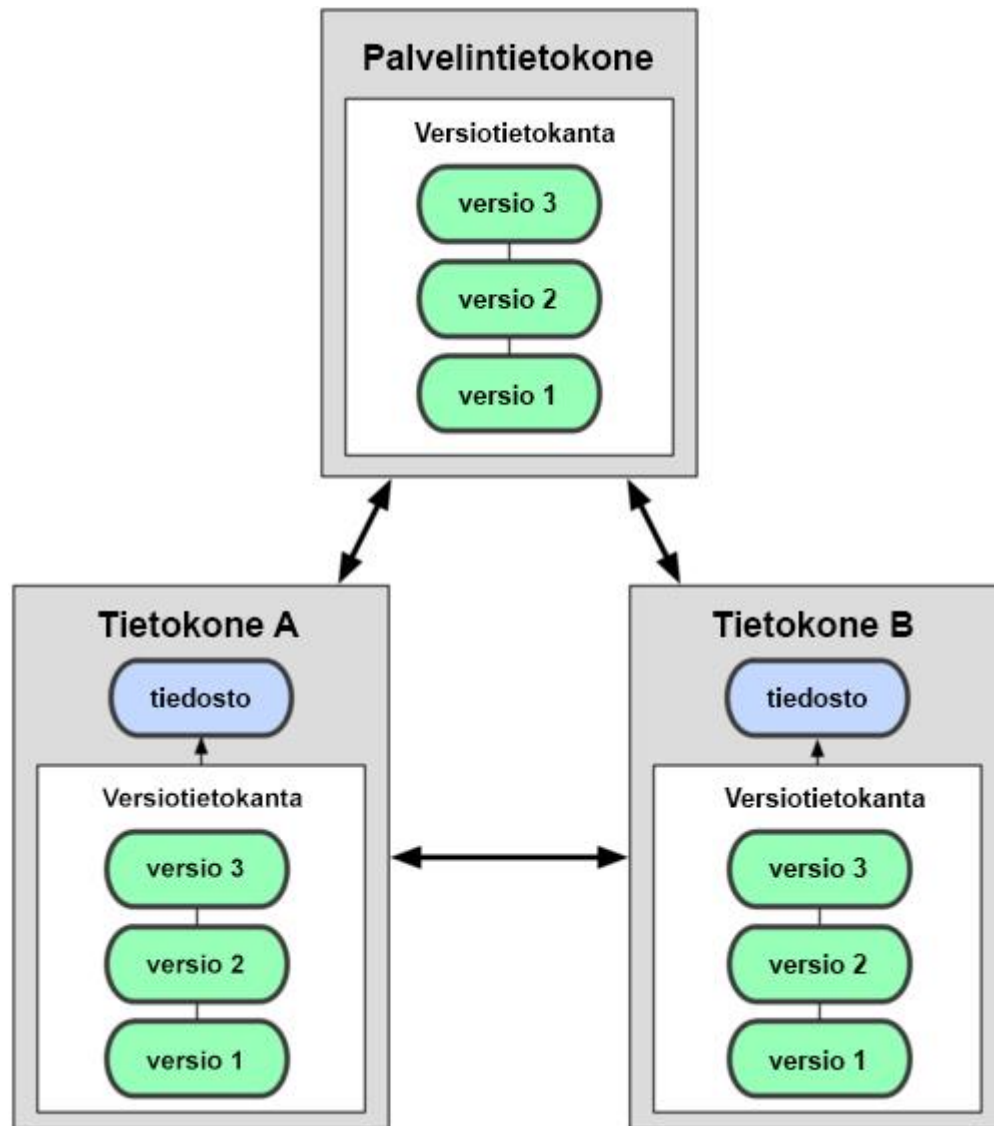
Molemmat tehtävänsuorittajat vaativat Node-palvelinympäristön toimiakseen. Less.js:n voi asentaa myös perinteisellä tavalla suoraan selainympäristöön, mutta sitä ei suositella, koska se hidastaa sivun latausta. Less joutuisi käsittelemään ja kääntämään tyyli-tiedostot, ennen kuin sivu voidaan näyttää.

3.4 Git-versionhallinta

Git on ilmainen avoimen lähdekoodin versionhallintaohjelmisto. Sen voi asentaa joko omalle palvelimelle tai käyttää verkossa olevia Git-palveluita, kuten GitHubia (<http://github.com/>) tai BitBucketia (<http://bitbucket.org/>).

Gitin ja yleensäkin versionhallinnan ajatuksena on säilyttää projektin vaiheiden eri versiot, jotta niihin voidaan tarvittaessa palata myöhemmin. Tästä on hyötyä varsinkin, kun esimerkiksi verkkosivukehittäjä tekee yhteistyötä muissa järjestelmissä olevien kehittäjien kanssa. Kehittäjä voi myös muokata sovellustaan isostikin tietäen, että aikaisempi versio on tallessa ja helposti palautettavissa. [Git 2015.]

Git on niin sanottu hajautettu versionhallintajärjestelmä (DVCM). Tämä tarkoittaa sitä, että palvelimella olevat tiedostot peilataan käyttäjien koneille niiden tehdessä tiedonhaun. Koko tietolähde on tällöin tallessa sekä palvelimella että käyttäjien koneilla. Se mahdollistaa myös useiden ihmisten ja ihmisryhmien osallistumisen samaan projektiin, kun kaikki voivat tehdä muutoksia lokaalisti omaan kopioonsa (kuva 4).



Kuva 4. Git, hajautetun versionhallintajärjestelmän malli [Git 2015].

Vaikka ei käyttäisikään versionhallintaa omassa projektissaan, Gitiin on silti miltei mahdollonta olla törmäämättä. Nykyään tuntuu, että suurin osa avoimista JavaScript-lisäosista ladataan Githubin kautta. Se tarjoaa myös käyttäjille mahdollisuuden vaikuttaa lisäosan ajan tasalla pysymiseen ja virheiden löytämiseen.

Projektissa Gitiä käytettiin teemapohjan ja lisäosien lataamiseen. Valmistunut teema ladattiin myös Githubiin opettajien tarkasteltavaksi.

4 Verkkosivun toteutuksen työnkulku

4.1 Projektin aloitus

Insinööriyöprojektina oli verkkosivun uusiminen KUTKE ry. -nimiselle yhdistykselle. KUTKE ry:n virallinen nimi on Kuntoutuksen tutkimus- ja kehittämissyhdystys KUTKE ry., mutta yleisesti sitä kutsutaan tuttavallisesti nimellä Kutke.

Yhdistyksen verkkosivu oli pahasti vanhentunut niin ulkonäöllisesti kuin sisällöllisesti-kin, ja se haluttiin uudistaa vastaamaan nykypäivän standardeja. Näköpiirissä oli myös tuleva EFRR (European Forum for Research in Rehabilitation) -konferenssi, joka järjestetään vuonna 2015 yhdessä KUTKE:n kanssa Helsingissä. Tavoitteena oli saada verkkosivu ja koko verkkoilme tähän mennessä uudistettua.

Pidimme aloituspalaverin yhdessä Kutken edustajan **Juha Lepän** ja Metropolian edustajan yliopettaja **Harri Airaksisen** kanssa.

Vanhan sivuston analysointi

Yhdistyksen vanha sivu oli niin sanottu staattinen HTML-sivu ilman sisällönhallintajärjestelmää. Se oli toteutettu vaalealle taustalle vihreällä navigaatio- ja otsikkopalkilla, ja tekstien väreinä oli käytetty violettiä ja punaista. Sisältösivuja oli 21. Kuvasta 5 nähdään, että kaiken kaikkiaan yleisvaikutelma oli sekava.



Kuva 5. Yhdistyksen vanha sivu, jota lähdettiin kehittämään.

Yhdistyksellä oli myös erillinen **Kuntoutusblogi** *Blogger*-palvelussa, johon navigaati-ossa oleva linkki ohjautui. Myös yksi englanninkielinen sivu oli tehty kansainvälisiä luki-joita varten, mutta läheskään kaikkia sivuja ei ollut käännetty.

Kuvasta 5 nähdään, että vasemmassa yläkulmassa, missä yleensä pidetään yrityksen logoa, oli pieni kuusiaiheinen valokuva. Yhdistyksen logo oli näkyvillä vain etusivun sisältötekstin alareunassa. Tämä ratkaisu aiheutti hämmennystä ja vähensi logon merkitystä yhdistyksen identiteetissä.

Kuvasta nähdään myös, että verkkosivun väritys oli sekava eikä edustanut yritystä missään mielessä. Päänavigaatio sijaitsi vasemmassa sivupalkissa. Siinä oli 21 linkkiä,

mikä teki siitä todella vaikeaselkoisen luettavan. Usein päänavigaatiopalkissa käytetään vain viidestä seitsemään linkkiä. Se on vielä sen kokoinen ryhmä, joka ihmisillä on tapana hahmottaa ennen pyrkimystä erotella sitä alaryhmiin [Miller 2011: 174]. Alaryhmit voi toteuttaa navigaatioissa esimerkiksi drop-down-menetelmällä, mitä ei myöskään vanhalla sivulla ollut käytetty.

Kuvasta 6 nähdään, miten paljon selkeämmäksi navigaatiota saatiin yhdistämällä osioita ja poistamalla turhia sivuja.



Kuva 6. Vanhan ja uuden sivun navigaation rakenne.

Tavoitteet ja toiveet

Yhdistyksen toiveina uutta verkkosivua varten oli ulkoasun päivittäminen, verkkosivun itsenäinen päivitysmahdollisuus jatkossa sekä englanninkieliset käännökset joka sivulle.

Aloituspalaverissa läpikäytyjen asioiden pohjalta päädyttiin myös siihen, että siirretään Kuntoutusblogi Bloggerista uudelle sivulle, jotta kaikki sisältö olisi saman verkkotunnuksen alla. Samalla mietittiin MailChimp-palvelun (<http://mailchimp.com/>) käyttöönottoa jäsenlomakkeita ja uutiskirjeiden lähettämistä varten sekä sosiaalisen median palveluita, kuten Facebookia ja Twitteriä. Päädyttiin siihen, että luodaan kaikki kolme tiliä ja katsotaan jälkeenpäin, löytyykö yhdistyksestä innokkaita sosiaalisen median päivittäjiä.

Verkkosivun sisällön kartoitus, kokoaminen ja priorisointi

Verkkosivun sisältö koostuu viidestä osasta: etusivu, sisältösivut, blogi, artikkelit ja tapahtumat. Näistä etusivu ja sisältösivut ovat pääasiassa staattisia, ja niitä ei päivitetä kuin tarvittaessa. Blogisivu listaa dynaamisesti uusimmat artikkelit ja Tapahtumat-sivu uusimmat tapahtumat. Yksittäinen artikkeli tai tapahtuma aukeaa omaan sivuunsa.

Etusivulle koettiin tarpeelliseksi listata uusimmat artikkelit ja tapahtumat.

4.2 Visuaalinen ilme ja rakennesuunnitelma

Logon suunnittelu

Heti alussa huomattiin iso ongelma: yhdistyksen logoa ei löytynyt keltään. Oli olemassa ainoastaan vanhalla verkkosivulla ollut pikselöitynyt JPG-logo ja kirjekuoresta skannattu logo. Totesin, että on tarpeellista jäljentää näistä logoista kunnollinen vektoritiedosto, jota voi myöhemminkin käyttää verkko- ja printtijulkaisuissa.

Koska alkuperäistä logotyyppiäkään ei ollut tiedossa, päätettiin, että kehitetään siihen uusi logotyyppi logomerkin rinnalle. Tähän sain vapaat kädet.

Piirsin logomerkin uudelleen *Adobe Illustrator* -ohjelmalla. Sain sen mahdollisimman lähelle oikeaa käyttäen JPG-logoa mallina. Väriin otin talteen pipetti-työkalulla, joten se ei välttämättä ole juuri alkuperäinen väri, johon se on suunniteltu, mutta tarpeeksi lähellä kuitenkin. Logotyyppiä valitsin aluksi Banda Regular -fonttiperheen, jonka pienet kaarevat päätteet toivat eloa logoon. Kuvassa 7 on logon uusi versio.



Kuva 7. Kutken uusiksi piirretty logomerkki ja logotyyppi.

Verkkosivun rakennesuunnitelma ja ulkoasun suunnittelu

Projektin suunnittelu on moniosainen ja monialainen prosessi. Sen vaiheisiin kuuluu yleensä ainakin jossakin määrin tutkimustyötä, sisällön kartoitusta, rautalankamallien suunnittelua, käytettävyyskartoitusta, prototyyppien tekoa ja esimerkiksi konseptointia. Hyvästä projektisuunnittelusta on monenlaisia hyötyjä; se voi esimerkiksi helpottaa tiimin työskentelyä isommissa yrityksissä ja auttaa asiakasta saavuttamaan pitkäaikaisempia tavoitteita. [Miller 2011: 151.]

Vanhan sivun analyysin perusteella oli ainakin selvää, että sivurakennetta pitää uudistaa selkeämmäksi ja loogisemmaksi, navigaatiota pitää selkeyttää ja sivuilla pitää olla yhtenäisempi värimaailma.

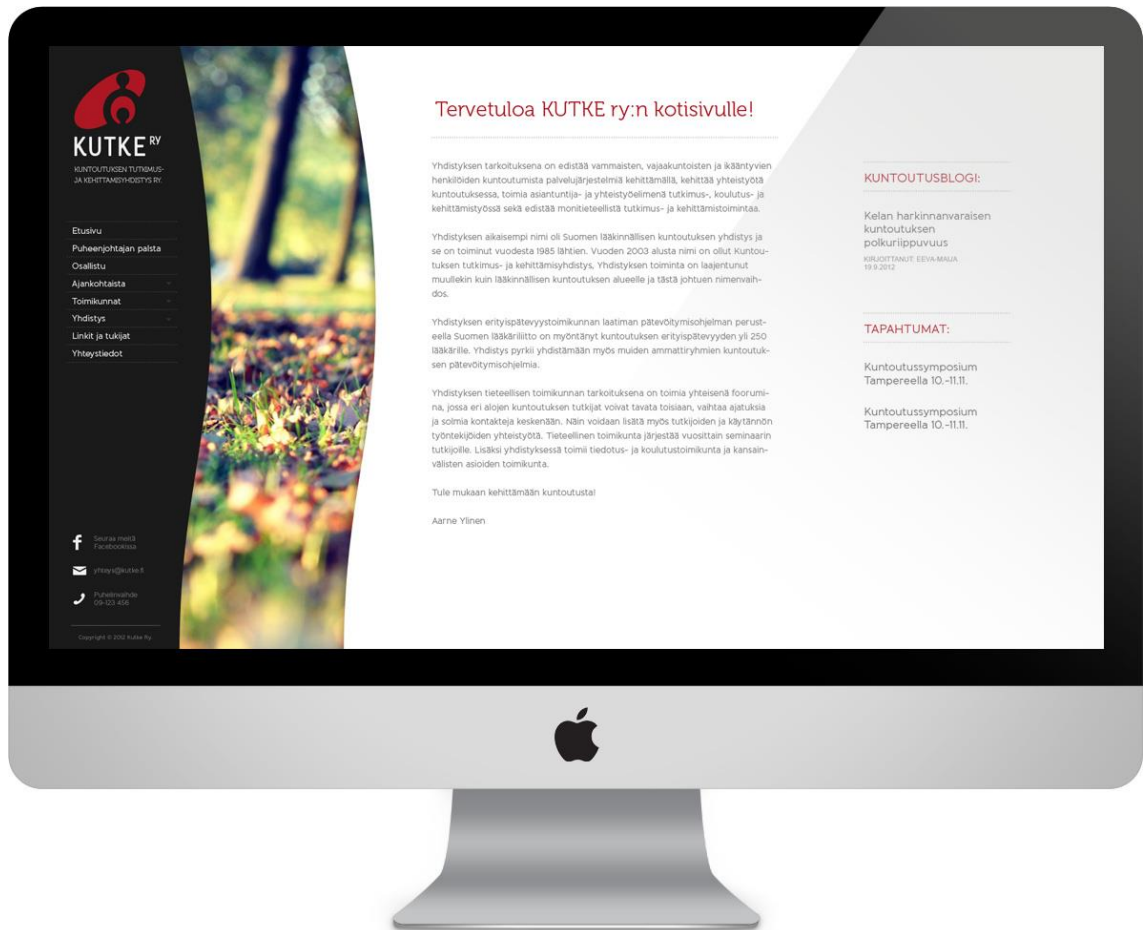
Aloitin miettimällä, mitä sisältöä sivuille tulee ja miten se kannattaa järjestää alasivuiksi. Uutta sisältöä ei blogin lisäksi tullut, joten järjestystä muuttamalla ja sivuja yhdistämällä saisi sisältöä jo selkeytettyä.

Koska verkkosivulla on suhteellisen paljon sivuja ja alasivuja, päätin pitää navigaation vasemmassa laidassa vanhan sivun tapaan. Navigaatioissa näkyvissä ovat vain pääta-son sivut, ja loput sivut löytyvät alavetovalikoiden alta. Alavetovalikot on merkitty nuolella, ja sitä klikkaamalla valikko avautuu. Sivupalkissa sijaitsevat myös linkit sosiaaliseen mediaan sekä kielenvaihtopainike.

Etusivulle oli asiakkaan puolelta tärkeää saada puheenjohtajan tervehdysteksti. Muita tarpeita ei ollut. Suunnittelin sivulle myös linkit uusiin artikkeleihin ja tapahtumiin. Yhdistyksen etusivulla olisi myös hyvä olla lyhyt kuvaus yhdistyksen toiminnasta ja mahdollinen sitaatti tai kuva, joka havainnollistaa yhdistyksen tavoitetta [Korpela 2005: 142]. Nykyään suositellaan myös *Call to action* -painiketta etusivulle heti kuvaustekstin jälkeen, mikä auttaa ohjaamaan kävijän oikeaan paikkaan – tässä tapauksessa hake-maan yhdistyksen jäseneksi.

Verkkosivu koostuu pääasiassa tekstisisällöstä ja analyttisistä artikkeleista. Kuvia ei juurikaan käytetä, joten visuaalisuus perustuu lähinnä väreihin, tunnelmakuvaan sekä typografiaan ja tekstin asetteluun. Koska brändin logossa oleva viininpunainen on niin voimakas, päätin käyttää sivuilla sen lisäksi vain mustaa ja valkoista. Ne jättävät pääpainon yhdistyksen päävärille ja samalla rauhoittavat sivua.

Kuvassa 8 on asiakkaalle lähetetty suunnitelma sivun ulkoasusta. Musta sivupalkki luo sivuille kontrastia, ja valokuva pehmentää siirtymistä valkoiselle sisältöalueelle. Valkoinen tasainen tausta tekstin alla parantaa luettavuutta ja luo tilan tuntua. Kaarevat rajapinnat luovat eloa muuten minimalistiseen näkymään ja tuovat siihen vähän logon pyöreyttä. Logon lisäksi brändiväri näkyy tehosteina otsikoissa ja linkeissä.



Kuva 8. Verkkosivun valmis ulkoasuunnitelmä.

4.3 Sivuston kehittämisen aloittaminen lokaalisti

Valitsin alustaksi Wordpress-sisällönhallintajärjestelmän, koska aikaisemman kokemuksen perusteella tiesin, että se sopisi tällaiseen projektiin. Siinä on myös sisäänrakennettu blogiominaisuus, mikä helpottaisi *Blogger*-blogin sisällön yhdistämistä verkkosivuihin.

Wordpressissä on myös mahdollista luoda useita käyttäjiä, mikä on otollista, koska blogiin kirjoittajia on monta. Kirjoittajat voivat halutessaan myös ladata Wordpress-mobiilisovelluksen puhelimiinsa tai tabletteihinsa, jolloin he pystyvät kirjoittamaan ja julkaisemaan myös liikkeellä ollessaan.

Wordpressin lokaali asennus vaatii, että koneelle on asennettu LAMP-kehitysympäristö, joka sisältää Apachen, MySQL:n ja PHP:n. Windowsille suosituin on XAMPP, jonka voi ladata osoitteesta <https://www.apachefriends.org/>. XAMPPin asennuksen jälkeen voidaan asentaa Wordpress, jonka voi myös ladata ilmaiseksi sen kotisivuilta. Jotta pääsee käyttämään Gruntia ja Lessiä, tulee asentaa myös Node.js. Nodesta on saatavilla Windows-asennustiedosto, jonka voi asentaa niin kuin minkä tahansa muunkin ohjelman.

Sivun runkona käytin Roots-teemasta aikaisemmin muokkaamaani Wordpress-teemapohjaa, joka on vain runko, joka sisältää tarvittavat tiedostot teemaa varten sekä Bootstrap-tyylitiedostot. Teemarungon kansioon asennetaan tarvittavat Grunt-moduulit ja aloitetaan kehitystyö.

4.4 Tehtävänsuorittajien ja esikäsittelijöiden käytön aloitus

Nodea ja sen lisäosia käytetään komentoriviltä. Jotta verkkosivun tai tässä tapauksessa Wordpress-teeman kehityksessä voitaisiin saada kaikki hyöty Nodesta ja sen pake-tinhallintasovelluksesta (NPM), tulee teemakansiossa olla tiedosto nimeltä *package.json*. Tässä tiedostossa määritetään moduulin tiedot ja tarvittavat komponentit, joita halutaan käyttää. Tiedoston voi myös luoda automaattisesti komentoriviltä komennolla *npm init*. Koodiesimerkissä 5 on projektityön *package.json*-tiedoston sisältö.

```
{
  "name": "roots",
  "version": "6.5.2",
  "author": "Janne Rättyä",
  "homepage": "http://jannerattya.com",
  "repository": {
    "type": "git",
    "url": "git://github.com/roots/roots.git"
  },
  "licenses": [{
    "type": "MIT",
    "url": "http://opensource.org/licenses/MIT"
  }],
  "engines": {
    "node": ">= 0.10.0"
  },
  "devDependencies": {
    "grunt": "~0.4.2",
    "grunt-contrib-clean": "~0.5.0",
    "grunt-contrib-jshint": "~0.6.4",
    "grunt-contrib-uglify": "~0.2.4",
    "grunt-contrib-watch": "~0.5.3",
    "grunt-contrib-less": "~0.8.1",
    "grunt-wp-version": "~0.1.0"
  }
}
```

Koodiesimerkki 5. *Package.json*-tiedoston sisältöä.

Kun kaikki on määritetty valmiiksi, voidaan komentoikkunassa suorittaa komento *npm install*, joka asentaa kaikki määritetyt komponentit teemakansioon. Esimerkkikuvassa on määritetty Grunt ja sen tarvitsemia moduuleita: *grunt-contrib-uglify*, *grunt-autoprefixer* ja niin edelleen.

Oman moduulin luomisesta on hyötyä varsinkin, kun käytetään versionhallintajärjestelmää, esimerkiksi Gitiä. Tällöin ei ole järkeä lähettää mukana kaikkia projektissa käytettyjä moduuleita.

tettyjä apukirjastoja ja koko *node_modules*-kansiota, sillä tarvittavat lisäosat saa ladata kätevästi komentoriviltä *npm install* -komennolla. Moduulien versioita voi myös seurata ja rajoittaa *package.json*-tiedoston avulla. Se tekee myös uudelleenjakamisen mahdolliseksi: oman moduulin voi jakaa muiden käytettäväksi NPM-rekisteriin komennolla *npm publish*. [Rauch 2012: 12.]

Grunt-tehtävänsuorittajan käyttö edellyttää myös asetustiedoston määrittämistä. Vaadittavat tiedostot *package.json*in lisäksi ovat *Gruntfile.js* tai *Gruntfile.coffee*, jossa määritetään suoritettavat tehtävät ja tiedostopolut. [Grunt: The JavaScript task runner 2015.]

Koodiesimerkissä 6 näkyy *gruntfile.js*-tiedoston sisältö. Esimerkissä on asennettu vain yksi lisäosa, *grunt-contrib-less*, ja jokainen käytettävä lisäosa tulee rekisteröidä erikseen. Less-lisäosalle tulee vähintään määrittää käännettävän tiedoston ja valmiin CSS-tiedoston nimi ja polut. Options-kohtaan voi määrittää muita lisäosan asetuksia. Asetusten määrittämisen jälkeen jokainen lisäosa ladataan erikseen `grunt.loadNpmTasks` -komennolla, ja lisäosat suoritetaan järjestyksessä `grunt.registerTask` -komennolla.

```
module.exports = function(grunt) {

// Projektin ja lisäosien konfiguraatio.
grunt.initConfig({
  less: {
    dist: {
      files: {
        'assets/css/main.min.css': ['assets/less/app.less']
      },
      options: {
        compress: true,
      }
    }
  }
});

// Lataa lisäosan, jossa less-tehtävä sijaitsee.
grunt.loadNpmTasks('grunt-contrib-less');

// Rekisteröi oletustehtävän/tehtävät.
grunt.registerTask('default', ['less']);

};
```

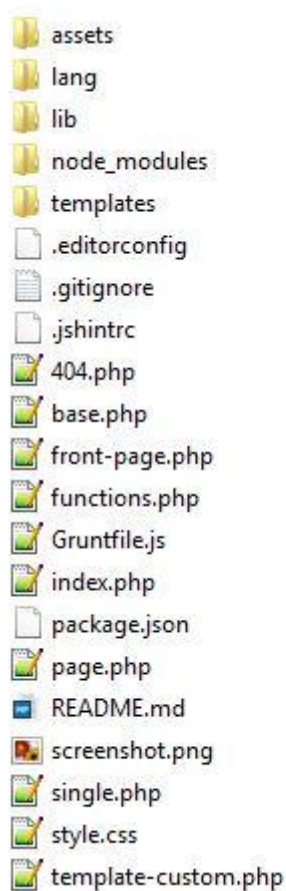
Koodiesimerkki 6. Gruntfile.js-tiedoston sisältö, jossa vain yksi lisäosa.

Käyttämässäni teemassa gruntfile.js määrittää seuraavat asiat:

- tarkistaa, että JavaScript-tiedostoissa ei ole virheitä
- tarkistaa, tarvitaanko CSS-tyyliin eteen selainkohtaisia etuliitteitä ja lisää ne
- muuttaa Less-tiedostot CSS-tiedostoksi ja pienentää sen sisällön
- yhdistää JavaScript-tiedostot yhdeksi tiedostoksi ja pienentää sen sisällön
- asettaa JavaScript- ja Less-tiedostot ja niiden kansiot tarkkailuun, ja jos jokin muuttuu, päivittää kaikki.

Kun tarvittavat Grunt-moduulit on asennettu *npm install* -komennolla, suoritetaan komentorivillä *grunt*-komento. Se suorittaa *gruntfile.js*-tiedostossa määritetyt tehtävät, minkä jälkeen voidaan laittaa päälle kansiontarkkailutila *grunt watch* -komennolla ja aloittaa teeman kehittäminen.

Kuvasta 9 nähdään teeman kansiorakenne. *node_modules*-kansio on Node lisäosien asennusta varten, eikä sitä kopioida teeman mukana palvelimelle tai versionhallintaan. *Assets*-kansiossa on alakansiot *css*, *fonts*, *img*, ja *less*. *Less*-kansion sisältö käännetään pienennetyksi CSS-tiedostoksi *css*-kansioon.



Kuva 9. Teeman kansiorakenne.

Teeman rakennus ja sisällön syöttö

Bootstrap-teeman rakenteen toteutus perustuu hyvin pitkälti gridiin eli ruudukkoon. Sisältöalue jakaantuu oletuksena 12 sarakkeeseen, ja divit järjestetään antamalla niille

tietty luokkanimi. Luokkanimet ovat muotoa *col-sm-6*, jossa numero tarkoittaa sarakkeiden leveyttä ja *sm* tarkoittaa *breaking pointia*, jonka jälkeen sarakkeveys astuu voimaan. Valmiit breaking pointit ja sarakkeveydet antavat mahdollisuuden luoda nopeasti monimutkaisiakin sivurakenteita erileveyksisille näyttöruuduille.

Bootstrapin ja monien muiden apukirjastojen toiminta perustuu siihen, että HTML-elementeille annetaan tietty luokkanimi, jolle on olemassa valmiiksi määritetty tyyli apukirjaston CSS-tiedostossa. Tällä tavoin sivusta saadaan nopeasti toimiva prototyyppi, minkä jälkeen sivua voi muokata ulkoasullisesti mieleiseseen CSS-tyyliä avulla.

Sivuille kopioitiin vanhan sivun sisältö, joitakin osioita poistettiin ja joitakin yhdistettiin. Seuraavana oli vuorossa artikkelien tuonti Blogger-palvelusta. Wordpressiin on saatavilla suoraan lisäosa nimeltään *Blogger Importer*, jonka avulla tuonti onnistui helposti. Lisäosa pyytää kirjautumaan Google-tilillä palveluun, minkä jälkeen voi valita omaan tiliin liitetyn Blogger-sivun, josta artikkelit tuodaan.

Artikkeleita varten haluttiin luoda Facebook-komentointimahdollisuus. Tätä varten tulee Facebookiin luoda oma App, jotta kommentointi voidaan autentikoida. Appin luomisen jälkeen saadaan yksilöllinen App ID, minkä jälkeen tarvittavat Facebook-lisäosat voidaan asentaa teeman sivupohjaan. Sivuille asennettiin kommentointikentän lisäksi tykkäysnappula.

Facebook- ja Twitter-jakamisen sujuvuuden takaamiseksi teeman *head*-osioon lisättiin myös sopivat *meta*-tagit. Tägeissa määritetään tietyt kuvaustekstit ja kuva, jotka halutaan näkyviin, kun artikkelin tai sivun linkki jaetaan palveluissa. Tähän on olemassa myös lisäosia, jotka tekevät tämän automaattisesti, mutta halusin tehdä sen käsin opiskelutarkoituksessa.

Sivuille asennettiin myös jäsenlomake jäseneksi liittyville. Tämä toteutettiin *Contact Form 7* -lisäosan avulla. Jäsenlomakkeen lisäksi sivulle lisättiin lomake uutiskirjeen tilaamiseksi. Uutiskirje toteutettiin MailChimp-palveluun (<http://mailchimp.com/>), johon myös tuotiin yrityksen jäsenrekisteri Excel-tiedostona. Toteutin yhdistyksen uutiskirjeen MailChimpin omalla editorilla, minkä jälkeen se lähetettiin kaikille yhdistyksen jäsenille (liite 1).

4.5 Julkaisu, testaus ja ohjeistus

Verkkosivua varten tilattiin web-hotellipalvelu Louhi.fi-yhtiöltä. Palvelu sisälsi kotisivutilan ja MySQL-tietokannan Wordpressin asennusta varten. Palvelun avulla on myös mahdollista luoda kutke.fi-päätteisiä sähköpostiosoitteita, mikäli tarvitsee.

Koska sivu luotiin ja kehitettiin lokaalisti, se piti seuraavaksi siirtää palvelimelle. Tämän voi tehdä monella tavalla. Jos sivulla on lokaalisti oma tietokanta, voi koko tietokannan ja Wordpress-asennuksen kopioida palvelimelle esimerkiksi phpMyAdmin- tai C-Panel-ohjelmien avulla. Tämän jälkeen joudutaan tosin käyttämään ulkopuolista ohjelmaa, esimerkiksi Search-Replace-DB (<https://github.com/interconnectit/Search-Replace-DB>), muuttamaan tietokannassa olevat lokaalin asennuksen polut uusiin.

Toinen vaihtoehto on asentaa palvelimelle puhdas Wordpress-asennus, kopioida wp-content-kansion sisällys (teema, lisäosat ja ladatut tiedostot) ja viedä sivun sisältö XML-tiedostona Wordpressin oman vienti- ja tuontityökalun avulla uudelle sivulle. Tässä on se ongelma, että sivun asetukset joutuu tekemään alusta uudelleen ja oletuskirjoitukset ja kategoriat poistamaan. Minulla on asennettuna lokaalisti samaan tietokantaan useita Wordpress-blogeja, minkä takia tämä vaihtoehto oli kuitenkin selkeästi helpompi.

Sivujen toimivuus testattiin Chromen, Firefoxin, Operan ja Internet Explorerin uusimmilla versioilla sekä matkapuhelimilla ja tableteilla. Sivusto löytyy opinnäytetyön kirjoittamisen aikaan kuitenkin osoitteesta <http://jannerattya.com/kutke>.

Tein Wordpressin käytöstä ohjeistuksen PDF-muodossa, ja sen avulla kuka tahansa uusi käyttäjä voidaan opastaa sivuja editoimaan. Ohjeistuksessa kerrotaan, miten lisääntään uusia artikkeleita ja muokataan vanhoja ja miten muokataan sisältösivuja ja lisääntään kuvia. Mielestäni on tärkeää opastaa sivun käyttö selkeällä tavalla, varsinkin jos asiakas on ATK-taidoiltaan rajoittunut.

4.6 Tulokset

Perinteisellä tavalla HTML- ja CSS-tiedostoja editoineelle Noden, Gruntin ja esikäsittelijöiden käyttö tuntuu hämmentävältä. Miksi nähdä kaikki se vaiva apuohjelmien asen-

nukseen ja tehtävänsuorittajien määrittelyyn, kun kärjitetysti sanottuna ilman siihen käytettyä aikaa sivu olisi jo valmistunut?

Syy liittyy optimaalisen työnkulun suunnitteluun verkkosivuja paljon työkseen tekeville. Vaikka oppimiskäyrä voi olla hidas, se maksaa itsensä takaisin työnkulun nopeutumisenä.

Joitakin havaitsemiani etuja Gruntin ja Lessin käytöstä:

- Syntaksi- ja kirjoitusvirheiden tarkistus tallennusvaiheessa helpottaa virheenkorjausta.
- Pakatut JavaScript- ja CSS-tiedostot pienentävät sivunlatausaikaa.
- Automaattinen selainkohtaisten etuliitteiden lisäys CSS-tyyleihin parantaa sivun selainyhteensopivuutta ja nopeuttaa sivukehitystä.
- Lessin muuttujat ja mixinit helpottavat toistuvien CSS-tyyliin kirjoittamista sekä mahdollistavat ”älykkäiden” tyylien luomisen.
- Lisäosia on lähes kaikkiin tarpeisiin.
- Maailmanlaajuisen kehittäjäyhteisön tuki takaa ajan tasalla pysymisen ja helpottaa ongelmien selvittämistä.

Havaitsemiani haittoja:

- Toiminnan opetteluun kuluu aikaa.
- Jos joku toinen kehittäjä haluaa myöhemmin jatkaa sivuston muokkaamista, tulee myös hänen osata Noden, Gruntin ja Lessin käyttö.
- Bootstrapin mukana tulee paljon ylimääräistä CSS- ja JavaScript-koodia, jota joutuu poistelemaan.

Itse projektin aikataulu ei pysynyt kovin hyvin koossa. Sivusuunnitelma oli nopeasti valmis, mutta asiakkaan kanssa toimiminen oli hidasta ja vastauksia sai odotella pitkään. Kehitystyö oli myös hidasta, koska mielipiteet muuttuivat useasti sekä asiakkaalla että itselläni. Koska tein itse sekä sivusuunnitelman että itse kehitystyön, huomasin kehitysvaiheessa asioita, joita en suunnitteluvaiheessa osannut ottaa huomioon. Siksi lopputuote on vähän erinäköinen kuin sivun ulkoasusuunnitelma.

Lopputuote on kuitenkin onnistunut ja selkeämpi kokonaisuus kuin aikaisempi sivu. Sivuille saatiin yhdistettyä erillisessä palvelussa ollut kuntoutusblogi ja sosiaalisen median kanavia. Etusivusta saatiin johdattelevampi ja navigaatiosta yksinkertaisempi.

Tulevaisuus

Verkkosivu jätettiin asiakkaalle sellaisenaan, ja asiakas ohjeistettiin käyttämään sivua. Mitä asiakas tekee sillä ja saadaanko sinne lisää kävijöitä, jää nähtäväksi.

Mitä esikäsittelijöihin, JavaScript-tehtävänsuorittajiin ja muihin Node.js-lisäosiin tulee, niiden päivitysrytmi elää omaa elämäänsä. Tätä kirjoitettaessa käyttämäni Rootsteemarunko on päivittynyt uuteen versioon, vaihtanut nimeä, siirtynyt käyttämään Gruntin sijasta Gulpia, ottanut käyttöön Bower-paketinhallinnan ja lisännyt tuen myös Sass-esikäsittelijälle.

Kehittäjälle voi tuottaa vaikeuksia pysyä uusien lisäosien ja päivitysten perässä ja päivittää jatkuvasti työnkulkuaan. Olennaisempaa onkin ehkä se, että löytää itselleen sopivan työnkulun ja JavaScript-apuvälineet ja pitäytyy niissä, kunnes jokin toinen tapa tuo jotain uutta merkittävää lisäetua.

5 Yhteenveto

Insinööriyössä toteutettiin Kutke ry -nimiselle yhdistykselle uusi verkkosivu ja verkkoilme sekä luotiin sosiaalisen median profiilit. Verkkosivu toteutettiin Wordpress-sisällönhallintajärjestelmään, joka asennettiin asiakkaan palvelimelle. Sivuuun yhdistettiin myös aikaisemmin erillisenä ollut blogi Blogger-palvelusta, luotiin MailChimp- uutiskirjepohja ja jäsenrekisteri sekä lisättiin Facebook-komentointi- ja jakolisäosat. Suunnittelin yhdistykselle myös uuden logon ja värimaailman käytettäväksi verkkosivuilla, uutiskirjeessä ja sosiaalisessa mediassa.

Tarkastelin nykyaikaisen verkkosivun vaatimuksia ja suositeltuja ominaisuuksia. Verkkosivulla tulee olla yhtenäinen visuaalinen ilme, ja sen tulee toimia tietokoneiden lisäksi pienillä mobiilinäytöillä. Tärkeässä osassa ovat myös sivun sisältö ja käytettävyys. Hyvän verkkoläsnäolon suunnittelun avulla käyttäjä löytää sivuille, vaikka ei tietäisikään oikeaa linkkiä.

Tarkastelin Wordpress-sivun suunnittelu- ja luontiprosessia suosittujen ja yhä yleistyvien JavaScript-apuvälineiden avulla. Teeman luomisen apuna käytin Node.js-suoritusympäristöä ja sen lisäosaa Grunt.js-tehtävänsuorittajaa. Projektissa käytettiin myös CSS-tiedostojen Less-esikäsittelijää sekä jQuery- ja Bootstrap-apukirjastoja.

Nämä apuvälineet on suunniteltu nopeuttamaan verkkosivun kehitystyötä automatisoimalla useasti toistuvia tehtäviä. Bootstrap-apukirjasto sisältää tyyli- ja JavaScript-moduuleita, joiden avulla suunnitellusta sivusta saadaan nopeasti toimiva prototyyppi ja runkomalli. Less-esikäsittelijä tehostaa CSS-tyyliä kirjoittamista tuomalla siihen muuttujia ja mixin-yhdistelmätyylejä. Grunt-tehtävänsuorittaja automatisoi erikseen määritettäviä tehtäviä. Sen avulla voi esimerkiksi oikolukea verkkosivun kehitysvaiheen tiedostoja, pakata CSS- ja JavaScript-tiedostot pienemmiksi ja lisätä selainkohtaisia etuliitteitä CSS-tyyleihin. Kaikki edellä mainitut tapahtuvat lokaalisti asennetussa Node.js- JavaScript-suoritusympäristössä.

Oikein käytettynä edellä mainitut apuvälineet voivat nopeuttaa työnkulkua merkittävästi, mutta käytön aloittamisen opettelu voi olla hidasta.

Lähteet

Coyer, Chris. 2007. Accessibility Basics: How Does Your Page Look To A Screen Reader? Verkkodokumentti. <<https://css-tricks.com/accessibility-basics-turn-your-css-off/>> 29.10.2007. Luettu 1.3.2015.

CSS3 @media Rule. Verkkodokumentti. Refsnes Data. <http://www.w3schools.com/cssref/css3_pr_mediaquery.asp> Luettu 1.3.2015.

Getting started | less.js. 2015. Verkkosivusto. Less Core Team. <<http://lesscss.org/>> Luettu 12.2.2015.

Git. 2015. Verkkosivu. Software Freedom Conservancy. <<http://git-scm.com/>> Luettu 16.2.2015.

Grunt – The JavaScript task runner. 2015. Verkkosivusto. <<http://gruntjs.com/>> Luettu 12.2.2015.

Gulp.js – The streaming build system. 2015. Verkkosivusto. Fractal. <<http://gulpjs.com/>> Luettu 12.2.2015.

Howe, Shay. 2014. Learn to Code Advanced HTML & CSS: Lesson 5 – Preprocessors. Verkkodokumentti. <<http://learn.shayhowe.com/advanced-html-css/preprocessors/>> Luettu 10.3.2015.

JavaScript technologies overview. Verkkodokumentti. Mozilla Developer Network. <https://developer.mozilla.org/en-US/docs/Web/JavaScript/JavaScript_technologies_overview> Päivitetty 2.4.2015. Luettu 9.4.2015.

Kliehm, Martin. 2007. Accessible Web 2.0 Applications with WAI-ARIA. Verkkodokumentti. <<http://alistapart.com/article/waiaria>> 9.4.2007. Luettu 1.3.2015.

Korpela, Jukka K. 2005. Web-suunnittelu. Jyväskylä: Docendo.

Korpela, Jukka K. 2011. HTML5 – Uudet ominaisuudet. Jyväskylä: WSOYpro.

Krug, Steve. 2006. Don't Make Me Think! A Common Sense Approach to Web Usability. Berkeley: New Riders Publishing.

Lawrence, Dave & Tavakol, Soheyla. 2007. Balanced Website Design – Optimising Aesthetics, Usability and Purpose. London: Springer.

Max, Stephan. 2014. An Introduction to WAI-ARIA. Verkkodokumentti. <<http://www.sitepoint.com/introduction-wai-aria/>> 21.7.2014. Luettu 1.3.2015.

- Marcotte, Ethan. 2011. Responsive Web Design. New York: A Book Apart.
- Miller, Brian. 2011. Above the Fold: Understanding the Principles of Successful Web Site Design. Ohio: HOW Books.
- Node.js. 2015. Verkkosivusto. Joyent. <<http://nodejs.org/>> Luettu 12.2.2015.
- Peltomäki, Juha. 2004. JavaScript. Jyväskylä: Docendo Finland.
- Purewal, Semmy. 2014. Learning Web App Development. Sebastopol, CA: O'Reilly Media.
- Rauch, Guillermo. 2012. Smashing Node.js – JavaScript Everywhere. Chichester: Wiley.
- Sani, Ilari. 2010. HTML5:stä rakentuu webin tulevaisuus. Verkkodokumentti. Micro PC. <<http://mpc.fi/nettilehti/pdf/1908201032.pdf>> Luettu 14.2.2015.
- StatCounter Global Stats. 2015. Verkkosivusto. StatCounter. <<http://gs.statcounter.com/>> Luettu 3.2.2015.
- W3C Recommendation - HTML5. 2014. Verkkodokumentti. W3C. <<http://www.w3.org/TR/html5/>> Luettu 14.2.2015.
- W3C – CSS3 Introduction. Verkkodokumentti. Refsnes Data. <http://www.w3schools.com/css/css3_intro.asp> Luettu 14.2.2015.

MailChimp- uutiskirjepohja

KUTKE ry:n jäsenkirje, Helmikuu 2014 [View this email in your browser](#)



*Kuntoutuksen tutkimus- ja kehittämissyhdystys KUTKE ry:n jäsenkirje
Helmikuu 2014*

Hyvät yhdistyksen jäsenet

Vuosi 2014 on jo hyvässä vauhdissa ja elämme sydäntalven aikaa, vaikka sitä ei eteläisessä Suomessa oikein talveksi tunnistakaan. Yhdistyksemme syyskokous joulukuun alussa 2013 kokoontui harvalukuisena mutta päättävaisena. Hallituksessa tapahtui muutamia henkilövaihtoksia. Allekirjoittanut jatkaa puheenjohtajana ja Tiina Leppinen sihteerinä. Muuta hallituksen jäsenet ja varajäsenet ovat Sinikka Peurala (varapuheenjohtaja), Heidi Anttila, Aarne Ylinen, Erja Poutainen, Esa Mäkinieni, Salla Sipari, Leena Penttinen, Eeva Leino, Juha Leppä, Ilona Autti-Rämö, Tuomas Leinonen ja Satu Luoto. Toimintasuunnitelma vuodelle 2014 hyväksyttiin ja se on tämän jäsenkirjeen liitteenä.

Vuosi 2014 tulee olemaan kuntoutuksen parissa työskenteleville monessa suhteessa merkittävä. Useita kuntoutusta suoraan koskevia ja sitä sivuavia lakiuudistuksia on vireillä. Kunnat ja kuntayhtymät ovat saaneet lausua mielipiteensä Sosiaali- ja terveydenhuollon palvelurakennemuutoksen lakiesityksestä. Tässä Sote-uudistuksessa on vielä monta allekirjoitaneelle avoimeksi jäänyttä kohtaa. Julkisen terveydenhuollon kuntoutuksen järjestämistä ajatellen lähtökohta on ollut ilman muuta positiivinen. On ajateltu, että kauan kaivattu kuntoutuksen alueellinen koordinaatio paranee ja resurssit saadaan optimaaliseen käyttöön. Palveluketjut tulevat kuin itsestään sujuviksi. Integraatiosta niin vertikaalisesti kuin horisontaalisestikin tulee totta. Kuinkahan käy? Jos Sote-alueiden sisälle synnytetään perustason itsenäisiä yksiköitä (joissa on vielä jonkinasteista erikoissairaanhoidoakin), saattaa tämä alueellinen koordinaatio jäädä haaveeksi. Entä hallinto? Isäntäkunnatko pelkäävät Sote-alueita johtavat? Näinköhän sillä demokratia kohenee? Mikä lienee erityisvastuualueiden (ERVA) rooli yliopistosairaaloiden kypeessä? Sote-lainsaadanto ei ole ainoa kuntoutusta koskeva mietinnän alla oleva lakiasia. Kelan ammatillisen kuntoutuksen ohjeistus tulee todennäköisesti muuttumaan. Samoin muutoksia tulee Työvoimahallinnon osatyökykyisten työnhakijoiden palveluihin. Työvoiman palvelukeskukset (TYP) saavat keskeisen roolin kaikkien työikäisten osatyökykyisten palvelujen järjestäjänä. Tämä koskettaa myös Kuntoutuksen asiakasyhteistyöryhmien toimintaa.

Yhdistyksen **kevätkokous** pidetään Kuntoutuspäivien yhteydessä, jälkimmäisen kongressipäivän päätteeksi **11.3.2014**. Varatkaa jo aika kalenteriini! Kokouskutsu asialistoineen tulee pian sähköposteihinne.


Sitten se yhdistyksen jäsenyyden velvoite: jäsenmaksu. Syyskokous piti jäsenmaksut ennallaan. **Vuoden 2014 Kute ry:n jäsenmaksu on 25 euroa ilman Kuntoutus-lehteä. Maksu on 70 euroa, jos samalla tilaa Kuntoutus-lehden.** Yhteisöjen jäsenmaksu on edelleen **150 euroa**.

Maksu maaliskuun puoliväliin mennessä tilille KUTKE ry Tampereen Seudun Osuuspankki, tilinumero **F149 5732 0920 0756 98**. Merkitkää viesti/viite-kohtaan: **Jäsenmaksu 2014 ja maksajan nimi**. Kuntoutus-lehti alkaa tippua postilaatikkoosi, kun maksu on saatu. Siksi pikainen jäsenmaksun maksaja saa lehdenkin jo ensimmäisestä numerosta alkaen.

Hyviä "talvi"-lomia kaikille, joilla sellainen mahdollisuus on! Toivottavasti kevätkokoukseen pääsee mahdollisimman moni.

*Heikki Suoyrjö
puheenjohtaja*

Liite: [KUTKE ry:n toimintasuunnitelma 2014](#)

 Facebook  Twitter  kutke.fi

Copyright © 2015 KUTKE ry. All rights reserved.
Sää tämäntoimittajan kooka kuka KUTKE ry:n jäsenrekisterin.

[unsubscribe from this list](#) [update subscription preferences](#)

MailChimp

Valmiin teeman kansiorakenne ja tarvittavat tiedostot

assets

css

main.min.css

fonts

img

js

vendor

jquery-1.11.0.min.js

modernizr-2.7.0.min.js

scripts.min.js

lang

fi_FI.mo

fi_FI.po

lib

activation.php

cleanup.php

comments.php

config.php

custom.php

gallery.php

init.php

nav.php

relative-urls.php

scripts.php

sidebar.php

itles.php

utils.php

widgets.php

wrapper.php

templates

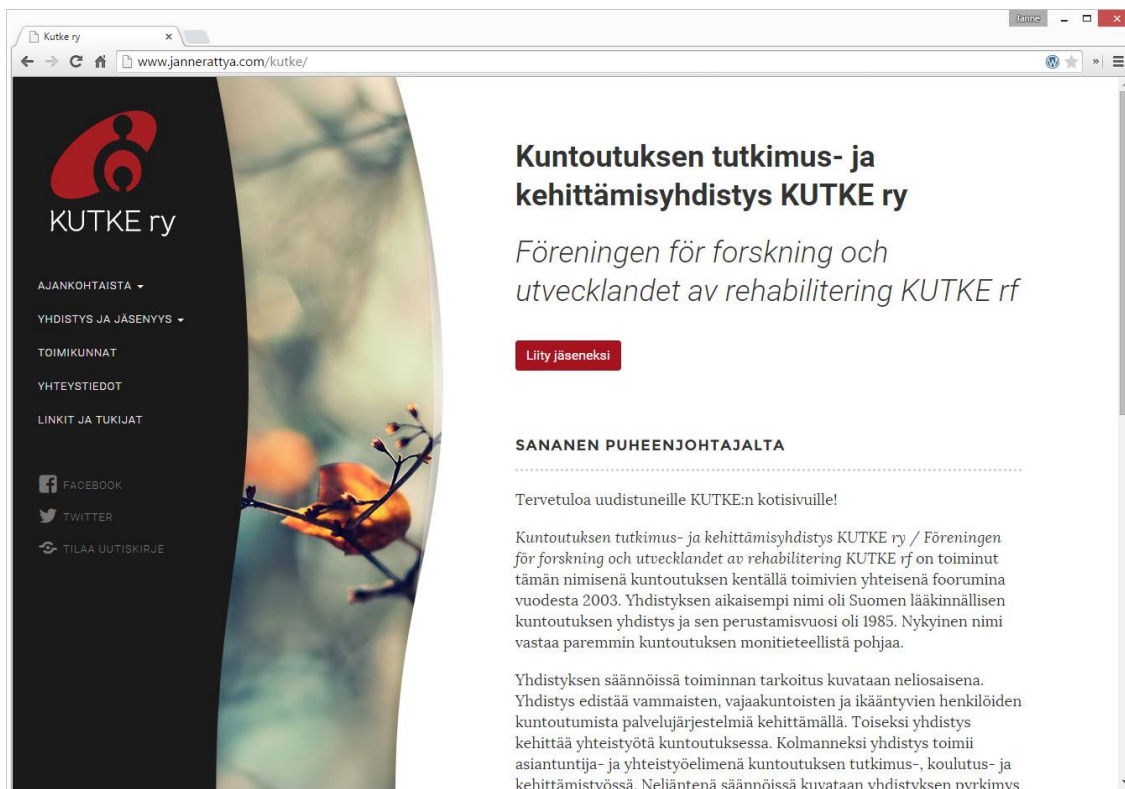
comment.php

comments.php


content.php

content-page.php
content-single.php
entry.meta.php
footer.php
head.php
header.php
header-top-navbar.php
page-header.php
searchform.php
sidebar.php
404.php
base.php
front-page.php
functions.php
index.php
page.php
screenshot.png
single.php
style.css




Valmis sivusto



Kutke ry
www.jannerattya.com/kutke/


KUTKE ry

AJANKOHTAISTA ▾
YHDISTYS JA JÄSENYYS ▾
TOIMIKUNNAT
YHTEYSTIEDOT
LINKIT JA TUKIJAT

 FACEBOOK
 TWITTER
 TILAA UUTISKIRJE

Kuntoutuksen tutkimus- ja kehittämissyhdistys KUTKE ry

Föreningen för forskning och utvecklandet av rehabilitering KUTKE rf

[Liity jäseneksi](#)

SANANEN PUHEENJOHTAJALTA

Tervetuloa uudistuneille KUTKE:n kotisivuille!

Kuntoutuksen tutkimus- ja kehittämissyhdistys KUTKE ry / Föreningen för forskning och utvecklandet av rehabilitering KUTKE rf on toiminut tämän nimisenä kuntoutuksen kentällä toimivien yhteisenä foorumina vuodesta 2003. Yhdistyksen aikaisempi nimi oli Suomen lääkinnällisen kuntoutuksen yhdistys ja sen perustamisvuosi oli 1985. Nykyinen nimi vastaa paremmin kuntoutuksen monitieteellistä pohjaa.

Yhdistyksen säännöissä toiminnan tarkoitus kuvataan neliosaisena. Yhdistys edistää vammaisten, vajaakuntoisten ja ikääntyvien henkilöiden kuntoutumista palvelujärjestelmiä kehittämällä. Toiseksi yhdistys kehittää yhteistyötä kuntoutuksessa. Kolmanneksi yhdistys toimii asiantuntija- ja yhteistyöelimenä kuntoutuksen tutkimus-, koulutus- ja kehittämistöissä. Neljäntenä säännöissä kuvataan yhdistyksen nvrkimvs

Kuntoutusblogi | Kutke ry x

www.jannerattya.com/kutke/ajankohtaista/kuntoutusblogi/

KUTKE ry

AJANKOHTAISTA ▾

YHDISTYS JA JÄSENYYS ▾

TOIMIKUNNAT

YHTEYSTIEDOT

LINKIT JA TUKIJAT

FACEBOOK

TWITTER

TILAA UUTISKIRJE

Kuntoutusblogi

Search Kutke

KATEGORIAT

Kuntoutusblogi

Taphtumat

KUTKE RY

Tule mukaan kuntoutuksen kehittämiseen ja yhteistyöhön!

Lity jäseneksi

EFFR 2015

8.5.2015 By Janne

Olemme mukana EFFR 2015 tapahtumassa. Lue lisää täältä.

Kelan harkinnanvaraisen kuntoutuksen polkuriippuvuus

16.5.2012 By Janne

Kela on tehnyt jo parikymmentä kolmivuotissuunnitelma Kelan kuntoutuslain 12 §:n mukaisesta budjettimäärärahasta (vanhassa vuoden 1991 laissa 4 §). Ensimmäinen suunnitelma ... Lue lisää

Kuntoutuksen käypä hoito

1.4.2012 By Janne

Duodecim-seura ja Kela ovat tehneet maaliskuussa kolmivuotisen (2012-2015) sopimuksen kuntoutusta koskevan tiedon saattamisesta ajan tasalle eri sairauksien ja vammojen hoidosta ... Lue lisää

Mitä kuntoutuksesta näkee

29.2.2012 By Janne

Kuntoutuksen käypä hoito: x

www.jannerattya.com/kutke/kuntoutuksen-kaypa-hoito/

KUTKE ry

AJANKOHTAISTA ▾

YHDISTYS JA JÄSENYYS ▾

TOIMIKUNNAT

YHTEYSTIEDOT

LINKIT JA TUKIJAT

FACEBOOK

TWITTER

TILAA UUTISKIRJE



Kuntoutuksen käypä hoito

1.4.2012 By Janne

Duodecim-seura ja Kela ovat tehneet maaliskuussa kolmivuotisen (2012-2015) sopimuksen kuntoutusta koskevan tiedon saattamisesta ajan tasalle eri sairauksien ja vammojen hoidosta annetuissa Käypä hoito -suosituksissa. Julkisuuteen annettun tiedotteen mukaan kuntoutuksen asemaa nostetaan hoidon ja diagnostiikan varjosta sille tasolle, jolle se tutkimustiedon valossa kuuluukin. Tähän mennessä kuntoutuksen kohtalona on ollut jäädä sivuosaan.

Kuntoutus-käypä hoitohankkeen yhteistyötahoja ovat THL:n, Kelan, Duodecimin ja erikoislääkäriyhdistysten lisäksi Fimea, Finska Läkaresällskapet ja Suomen Fysioterapeutit. Hankkeen tavoitteena on kehittää menettelytavat, joilla varmistetaan kuntoutuksen asema osana hoitosuosituksia. Sisältötyö rytmittyy sen mukaan kuin uusia suosituksia otetaan laadittaviksi ja entisiä päivitetään kolmen, viiden vuoden sisällä. Nämä päivitykset huomioon ottaen vanhimmat suositukset ovat nyt vuodelta 2005. Kokonaiskatselmuksen kuntoutuksen asemasta voisi tämä hanke tehdä, vaikka itse Käypä hoito-toimitus ei siihen ryhtyisikään. Koska tarkoitus on myös kuulla (ja aktivoita?) useita kuntoutuksen ammattiryhmiä ja potilasjärjestöjä, niin kaikki suositukset voivat joutua tarkastelun alle.

Duodecim käynnisti Käypä hoito-suositusten kehittämisen vuonna 1994 ja ensimmäisenä ilmestyi keliakia-suositus 1997. Vuosi sitten toukokuussa suositusten määrä ylitti jo sadan. Alusta pitäen ovat kuntoutusalan ihmiset varmaankin tarkastelleet näitä suosituksia myös omalta kannaltaan, ja pohtineet kuntoutusta koskevan tiedon ja ohjeistuksen riittävyyttä. Eteneminen on ollut hyvin nihkeää, ja Kelan



Kelan harkinnanvaraisen kuntoutuksen polkuriippuvuus

16.5.2012 By Janne

Kela on tehnyt jo parikymmentä kolmivuotissuunnitelma Kelan kuntoutuslain 12 §:n mukaisesta budjettimäärärahasta (vanhassa vuoden 1991 laissa 4 §). Ensimmäinen suunnitelma koski vuosia 1993-1995 ja suunnitelma oli viisi sivua. Kelan hallitus päättää asiasta helmikuussa ja toimittaa suunnitelman välittömästi STM:lle, jonka vakuutusosasto ottaa asian käsittelynsä ja pyytää Kuntoutusasiain neuvottelukunnalta lausunnon. Tänä vuonna ajankohdat olivat asiakirjamerkintöjen mukaan: Kelan hallituksen päätös 3.2.2012, lähettäminen STM:ään 3.2.2012 ja vakuutusosaston lausuntopyyntö KUNKilta 3.2.2012. Kukaan ei voi syyttää valtion elimiä hidastelusta. KUNKilta meni aikaa enemmän, kun kokouksia ei ole joka päivä, ja asia käsiteltiin 13.4.2012 pidetyssä kokouksessa.

Kelan kuntoutuslaki 1991 4 §

Muu ammatillinen ja lääkinnällinen kuntoutus

Kansaneläkelaitoksen on vuosittain laadittava 1 momentissa tarkoitettujen varojen käytöstä kolme seuraavaa kalenterivuotta koskeva suunnitelma sen mukaan kuin asetuksella tarkemmin säädetään.

Jos vuosittain yksilökohtaiseen harkinnanvaraiseen kuntoutukseen osoitetusta enimmäisrahamäärästä on osa jäänyt käyttämättä, se voidaan käyttää yksilökohtaiseen kuntoutukseen seuraavina vuosina.

- ammatillisesti ja/tai työyhteisöllisesti suuntautunut kuntoutus
- sairausryhmäkohtaiset kurssit
- yksilölliset kuntoutusjaksot
- muut kuntoutustoiminta

Yksi tapa jolla Kela reagoi kurssien osanotto-ongelmiin, on palata vanhaan malliin painottaen yksilöllisiä hoitajaksoja etukäteen määrättyjen kurssien asemesta. Tämän ohella tai ehkä painokkaammin voisi luoda toimivia yhteistyökäytäntöjä terveydenhuollon kanssa kummallekin osapuolelle (Kela ja kuntien terveydenhuolto) sopivien palvelujen ja palveluketjujen rakentamiseksi. Eikä kuntoutujia ja heidän omia tässä saisi unohtaa.

[Tykkää](#) [Jaa](#) Ole kavereistasi ensimmäinen, joka tykkää tästä.

Kommentoi Facebook-tunnuksellasi:

0 Comments Sort by Parhaat

 Add a comment...

Facebook Comments Plugin

© 2015 Kutke ry

