

Responsiivisen ulkoasupäivityksen toteuttaminen Plone-sivustolle

Ari Puttonen

Opinnäytetyö
Toukokuu 2015

Mediatekniikan koulutusohjelma
Tekniikan ja liikenteen ala





Tekijä(t) Puttonen, Ari	Julkaisun laji Opinnäytetyö	Päivämäärä 10.5.2015
	Sivumäärä 34	Julkaisun kieli Suomi
		Verkkojulkaisulupa myönnetty: x
Työn nimi Responsiivisen ulkoasupäivityksen toteuttaminen Plone-sivustolle		
Koulutusohjelma Mediatekniikan koulutusohjelma		
Työn ohjaaja(t) Pasi Manninen		
Toimeksiantaja(t) Jyväskylän yliopiston IT-palveluiden Kehittämispalvelut		
Tiivistelmä <p>Opinnäytetyön tavoitteena oli toteuttaa ulkoasupäivitys Each One Teach One -Plone-sivustolle Diazo-teemoitusrajapintaa hyödyntämällä. Työn toimeksiantajana toimi Jyväskylän yliopiston IT-palveluiden Kehittämispalvelut. Sivuston vanha ulkoasu on ollut käytössä jo useita vuosia, mutta sivusto on yhä aktiivisessa käytössä. Ulkoasu-uudistuksesta tuli ajankohtainen, sillä sivuston Plone-versio tullaan päivittämään syksyllä 2015.</p> <p>Työssä käsiteltiin Plonen ominaisuuksia yleisellä tasolla sekä Plonen teemoittamisen mahdollistavaa Diazo-tekniikkaa. Nykyään sivustojen tarvitsee olla toiminnallisuudeltaan päätelaitteesta riippumattomia, joten myös responsiivinen verkkosuunnittelu oli tärkeässä osassa. Tärkeintä oli ulkoasun päivittäminen helpommin ymmärrettäväksi Each One Teach One -sivuston loppukäyttäjiä, eli kurssin suorittajia varten.</p> <p>Ulkoasupäivitys toteutettiin Plone 4.3 -sisällönhallintajärjestelmää varten. Toteuttaminen aloitettiin luomalla responsiivinen sivustopohja, jonka responsiivisuus toteutettiin hyödyntäen CSS3-tyylittelyä sekä jQuery JavaScript-kirjastoa. Sivustopohja yhdistettiin Plonen sisällön kanssa Diazon teemoitusmoottorin tarjoamien XML-pohjaisten sääntöjen sekä XSLT-määrittelyjen avulla.</p> <p>Opinnäytetyön tuloksena saatiin Ploneen asennettava Diazoteema, joka päivittää Each One Teach One -sivuston ulkoasun responsiiviseksi sekä helpommin hahmotettavaksi kokonaisuudeksi. Ulkoasupäivitys sivustolle toteutetaan vasta syksyllä 2015, joten ulkoasun kehittäminen jatkuu opinnäytetyön päättymisen jälkeenkin.</p>		
Avainsanat (asiasanat) Web-kehitys, Responsiivinen verkkosuunnittelu, Plone, Diazo		
Muut tiedot		



Author(s) Puttonen, Ari	Type of publication Bachelor's thesis	Date 10.5.2015
		Language of publication: Finnish
	Number of pages 34	Permission for web publication: x
Title of publication Creating responsive appearance for Plone web site		
Degree programme Media Engineering		
Tutor(s) Manninen, Pasi		
Assigned by Development services of IT Services at University of Jyväskylä		
Abstract <p>The aim of this bachelor's thesis was to update the looks of Each One Teach One web site by using Diazo theming engine. The project was implemented in cooperation with Development Services of IT Services at University of Jyväskylä. The old appearance of the web page has been in use for many years, however, the site is still in active use. Creating a new appearance came in to the picture, because the Plone version of the web page will be updated in Autumn 2015.</p> <p>The work deals with Plone functionality in general and Diazo theming engine, which enabled theming Plone. Nowadays web pages need to work in all kinds of devices, therefore responsive web design was also an important part of the process. The most important part was updating the looks for Each One Teach One web pages' end users, the course participants.</p> <p>The updated looks were created for Plone 4.3 content management system. The process began with creating a responsive web page layout. CSS3 styling and jQuery JavaScript library were implemented for this purpose. The layout was merged with the content of Plone by using XML based rules and XSLT coding provided by Diazo theming engine.</p> <p>The end-product was a Plone theme made possible by Diazo, which updated the looks of Each One Teach One web page to be a responsive and more cohesive entity. The new Plone version will be updated for production in Autumn 2015, therefore the work with the new layout will continue further after this bachelor's thesis has been finished.</p>		
Keywords/tags (subjects) Web-development, responsive web design, Plone, Diazo		
Miscellaneous		

Sisältö

1	Työn lähtökohdat	3
1.1	Tausta ja toimeksiantaja	3
1.2	Tavoitteet ja tehtävät	3
2	Plone ja sen teemoitus	4
2.1	Sisällönhallintajärjestelmän tarkoitus	4
2.2	Plone	5
2.3	Deliverance	8
2.4	Diazo	8
3	Responsiivinen web-suunnittelu	10
3.1	Lähtökohta	10
3.2	Mukautuva ruudukko	11
3.3	Mukautuvat kuvat	12
3.4	Mediakyselyt	12
3.5	Nykyinen tilanne	13
3.6	Vakiintuneet käytännöt	13
3.7	Vanhentuvat käytännöt	14
3.8	Uudet tulokkaat	14
4	Each One Teach One -sivuston ulkoasu-uudistus	14
4.1	Mikä on EOTO?	14
4.2	Vanha ulkoasu	15
4.3	Uusi ulkoasu	18
4.4	Tyhjän pohjan luominen	19
4.5	Esimerkkejä teeman rakenteesta ja responsiivisuudesta	21
4.6	Plonen teemoitustyökalu	25
4.7	Ulkoasun implementointi Ploneen	27

5 Tulokset ja pohdinta.....	31
Lähteet.....	33

Kuviot

Kuvio 1. Plonen vakioteema	6
Kuvio 2. Plonen sisällönhallintapalkki	7
Kuvio 3. Plonen tavallinen ulkoasu vasemmalla, Diazo-teema oikealla	10
Kuvio 4. Esimerkki kuvien asettelusta eri kuvasuhteita tukeville näytöille	13
Kuvio 5. Vanhan sivuston kansiorakenne.....	15
Kuvio 6. Vanhan ulkoasun etusivu	16
Kuvio 7. Vanhan ulkoasun Ilmoitukset -sivu	16
Kuvio 8. Vanhan ulkoasun Blogin etusivu	17
Kuvio 9. Vanhan ulkoasun blogisivun sisältösivu	18
Kuvio 10. Uuden ulkoasudemon informaationsivu.....	19
Kuvio 11. Demon Blogisivu täysikokoisena vasemmalla, kapeana oikealla	25
Kuvio 12. Plonen teemoitustyökalu	26
Kuvio 13. Plonen teeman muokkausnäkyä	27
Kuvio 14. Sivuston teemoitettu etusivu	31

1 Työn lähtökohdat

1.1 Tausta ja toimeksiantaja

Opinnäytetyön toimeksiantajana toimi Jyväskylän yliopiston IT-palveluiden Kehittämispalvelut. Kehittämispalveluiden tehtävänä on kehittää, ylläpitää ja tukea IT-palveluissa kehitettyjä sovelluksia yliopiston opiskelijoiden ja henkilökunnan käyttöön. Osa sovelluksista on ulkoa hankittuja, mutta osa on kehitetty itse, usein käyttäen avoimen lähdekoodin ratkaisuja.

IT-palveluissa on tehty paljon sivustoja, sovelluksia ja palveluita Plone-sisällönhallintajärjestelmän päälle. Esimerkiksi Plonea on käytetty Jyväskylän yliopiston verkkosivuja (2 miljoonaa sivun latausta kuukausittain), Moniviestin-videojulkaisualustaa ja Koppa-kurssimateriaaliportaalia luodessa. Yksi näistä palveluista on Jyväskylän yliopiston kielikeskukselle Plonella toteutettu Each One Teach One -sivusto, jonka ulkoasu-uudistusta tämä opinnäytetyö käsittelee. Plone-kehityksessä työskentelee kolme vakituista työntekijää sekä muutama määräaikainen. He kehittävät myös muita Python-pohjaisia ohjelmointiratkaisuja.

1.2 Tavoitteet ja tehtävät

Opinnäytetyön tavoitteena oli toteuttaa uudistettu ulkoasu Jyväskylän yliopiston kielikeskuksen Each One Teach One (jatkossa EOTO) -verkkosivustolle. Samalla oli myös tarkoitus tutustua Plone-sisällönhallintajärjestelmään ja perehtyä tarkasti sekä responsiiviseen verkkosuunnitteluun että Diazo-teemoitukseen.

EOTO on kielikeskuksen ohjaama pari- tai ryhmätyöskentelyohjelma, jossa opiskelijat opettavat toisilleen omaa äidinkieltään tai koulukieltään ja opiskelevat parinsa kieltä. Opiskelijat tapaavat säännöllisesti vähintään kahdenkymmenen tunnin ajan täyttäen sivustolla olevaan blogiin raportit jokaisesta tapaamisesta. EOTO-sivuston toiminta käsitellään tarkemmin kappaleessa 4.1.

Kielikeskuksen EOTO-sovellus on ollut käytössä tuhansilla opiskelijoilla samanlaisessa muodossa jo vuodesta 2009 alkaen. Sovellusta on ylläpidetty pienin korjauksin ja tietoturvapäivityksin, mutta sen ulkoasuun tai käytettävyyteen ei ole tehty muutoksia. Sovellus toimii Plone 3 -version päällä, mutta tarvitsisi päivityksen Plone 4.3 -versioon, jotta se olisi jatkossa paremmin ylläpidettävissä ja kehitettävissä edelleen.

Versiopäivityksessä tulisi tehdä ulkoasu-uudistus, ja oletuksena versiopäivityksen yhteydessä on tarjolla yliopiston oletusteema. Kielikeskuksen intresseissä oli kuitenkin pitää tunnistettavuudesta kiinni.

Kehittämispalveluiden Plone osaajat ovat tehneet vain muutaman toteutuksen tässä opinnäytetyössä esiteltävällä Diazo-tekniikalla. Diazolla on tarkoitus tehdä jatkossa muun muassa yliopiston hae opiskelemaan ja opiskelu-sivustojen teema ja sen jälkeen koko Jyväskylän yliopiston verkkosivuston ulkoasu-uudistus.

Työn pääpaino oli verkkosivupohjan luomisessa ja sen muuntamisessa Plonen teemaksi Diazon tarjoamaa rajapintaa hyödyntäen. Diazo tarjosi mahdollisuuden yhdistää staattinen verkkosivupohja jo valmiiseen sisällönhallintajärjestelmällä toteutettuun sivustoon. Lisää tietoa Diazosta löytyy kappaleesta 2.4. Samalla oli tarkoituksena luoda ulkoasusta nykyaikainen päätelaitteesta riippumaton käyttökokemus, joten responsiivinen verkkosuunnittelu oli myös tärkeä osa-alue.

Opinnäytetyön motivaationa toimi halu syventää ymmärrystä verkkosivustojen kehittämisestä samalla tutustuen Ploneen sisällönhallintajärjestelmänä. Koska ulkoasu-uudistus toteutettiin Diazon avulla, sitä tehdessä ei kuitenkaan ollut tarve ryhtyä täysiveriseksi Plone-kehittäjäksi. Samalla oli mielenkiintoista saada kurkistaa Jyväskylän yliopiston IT-palveluiden Kehittämispalveluiden toimintatapoihin. Toisaalta tiimi sai arvokasta tietoa Diazo-tekniikalla kehittämisestä ulkopuolisen verkkokehittäjän silmin toteutettuna.

2 Plone ja sen teemoitus

2.1 Sisällönhallintajärjestelmän tarkoitus

Sisällönhallintajärjestelmä on järjestelmä, jonka avulla on mahdollista koostaa ja muokata sisältöä. Sisällönhallintajärjestelmiä käytetään useimmiten verkkosivujen ylläpitoon, mutta niitä voidaan hyödyntää myös yritysten sisäisten asioiden hoitoon. Tästä hyvä esimerkki on sisäisten viestien sekä ohjeiden jako pelkästään rajatulle käyttäjäryhmälle. Eri oikeuksien lisäksi sisällönhallintajärjestelmiin usein sisältyy laaja tuki erilaisille sisällöille, kuten kuville, videoille ja PDF-dokumenteille. Sisällönhallinta-

järjestelmää valittaessa onkin tärkeää ensin selvittää mitä ominaisuuksia siltä vaatii. (Luckow & Frank 2013.)

Opinnäytetyön tavoitteena ei ole keskittyä vertailemaan erilaisia sisällönhallintajärjestelmiä tai pureutua niiden tarkoitukseen tarkemmin. Huomion keskipisteenä sen sijaan on Plone-sisällönhallintajärjestelmä ja sen teemoitus.

2.2 Plone

Plone on helppokäyttöinen sekä erinomaisesti skaalautuva että turvallinen Pythoniin pohjautuva sisällönhallintajärjestelmä. Yksi Plonen suurimmista vahvuuksista on sen kypsyys. Sen kehittämisprojekti on jatkunut aktiivisena vuodesta 2002 lähtien. Hyvä esimerkki kehittämiseen käytetystä määrästä ovat Plonen kehittämiseen käytetyt työtunnit. Arviot projektiin upotetuista työtunneista kirjoitettujen koodirivien perusteella pyörivät 350 työvuoden paikkeilla. Plone on saatavilla neljäkymmenellä eri kielellä ja se on määritelty kaikkein turvallisimmaksi merkittäviin sisällönhallintajärjestelmiin sisältyvistä vaihtoehdoista. (About Plone 2015.)

Plone on turvallisempi kuin mikään PHP-pohjainen julkaisujärjestelmä. Esimerkkinä Plone julkaistiin vuonna 2003, mutta siitä huolimatta Plonesta on löytynyt vain alle viisikymmentä raportoitua heikkoutta. Verrattaessa esimerkiksi Drupaliin, WordPressiin tai Joomlaan, jokaisesta edellämainitusta sisällönhallintajärjestelmästä on löydetty yli kymmenkertainen määrä heikkouksia. (Mt.)

Plonen omistaa voittoa tavoittelematon Plone Foundation, jonka tavoitteena on varmistaa Plonen riippumattomuus yrityksistä tai yksityisistä henkilöistä. Plone yhteisö on laaja ja sisältää kehittäjiä noviiseista kovaksi koulittuihin koodaajiin. Plonea ovat käyttäneet muun muassa NASA, Nokia, Oxfam, Amnesty International, eBay ja Novell. (What is Plone? 2015.)

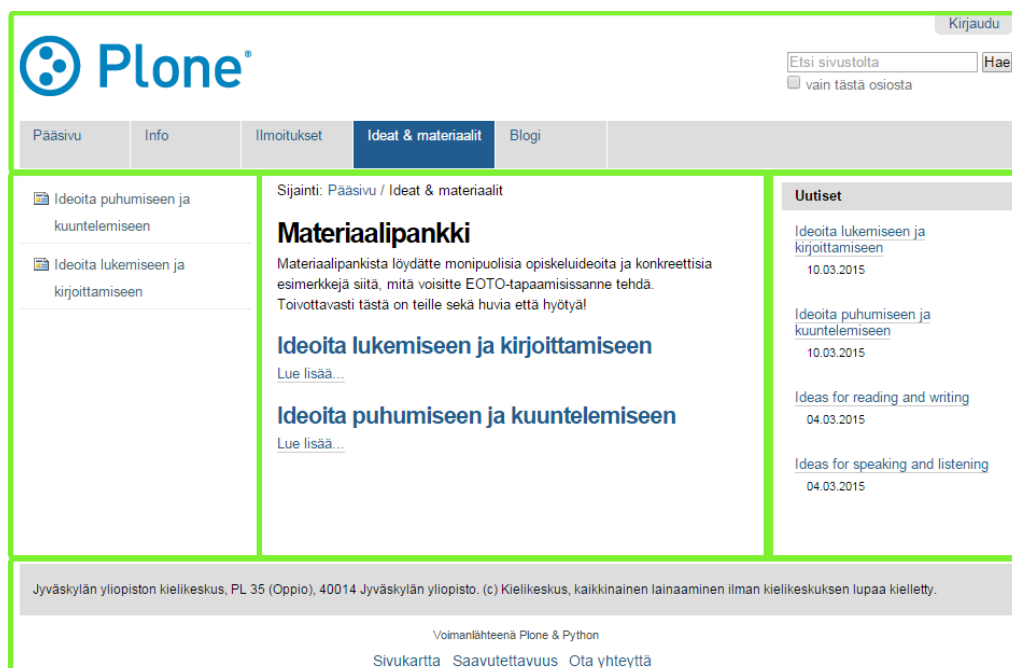
Plone sai nimensä saman nimiseltä elektronisen musiikin bändiltä. Plonen perustajat kuuntelivat Plone-bändiä tavatessaan (ja koodatessaan Plonea). Bändi hajosi vuonna 2001 mutta sisällönhallintajärjestelmä Plone jatkoi elämäänsä. (Mt.)

Viimeisin julkaistu vakaa versio Plonesta kirjoitushetkellä on 4.3.4. Omalle koneelle Plonen voi asentaa käyttämällä universaalaa asennusohjelmaa, joka toimii Linux- ja Unix-pohjaisilla käyttöjärjestelmillä. Windows pohjaisille käyttöjärjestelmille on saa-

tavilla karsittu asennusohjelma, jonka toiminnallisuus on vajaavainen verrattuna muiden järjestelmien versioihin. Sen sijaan suositeltu kehittämistapa Windows-järjestelmissä on asentaa Plone virtuaalikoneeseen. (Quickstart 2015.)

Opinnäytetyötä varten päädyttiin asentamaan Plone ilmaispalveluun nimeltä Heroku. Käyttäjä pääsee lisäämään testaamiseen tarvittavat lisäohjelmat Ploneen sekä tarvittaessa jopa tekemään siitä pienimuotoiseen tuotantoon kelpaavan palvelun. Näin myös työn etenemistä oli helpompi seurata, kun sisältö on nähtävissä verkossa.

Plonen vakio ulkoasu koostuu viidestä eri osasta: otsikosta, pääsarakeesta, vasemmasta sekä oikeasta sarakeesta ja sivun alaosa. (Ks. kuvio 1.) Otsikko-osa koostuu logosta, kirjautumistyökaluista ja navigaatiosta. Pääsarake sisältää sivun pääasiallisen sisällön sivulla olevien sarakkeiden keskittyessä lisäämään sivulle aputyökaluja, kuten esimerkiksi kalenterin tai kansiorakenteen linkkeineen. Alaosa on sivustoille tyypillistä alaviitettä varten. (Conceptual Overview 2015.)



Kuvio 1. Plonen vakioteema

Kirjautumattomat käyttäjät näkevät kaiken julkaistuksi määritellyn sisällön. Julkaisuun sisältöä sivuilla kirjautuneella käyttäjällä tulee olla oikeudet lisätä sivuston kyseessä olevaan osioon sisältöä. Oikeudet määritetään siis tarvittaessa kansio-, tai jopa tiedostokohtaisesti. Mikäli kirjautuneella käyttäjällä on tarvittava määrä oikeuk-

sia, päänavigaation alle ilmestyy vihreä sisällönhallintapalkki. (Plone User Accounts and Roles 2015.) (Ks. Kuvio 2.)



Kuvio 2. Plonen sisällönhallintapalkki

Sisältö-välilehti tuo esiin sen hetkisen kansion sisällön, Näytä-välilehti näyttää oletus-sisällön, joka näkyy kaikille, Muokkaa-välilehdeltä pääsee muokkaamaan sivun sisältöä ja Käyttöoikeudet-välilehdeltä pääsee asettamaan käyttäjäkohtaisia oikeuksia sivulle. (Mt.)

Plone tukee suoraan asennuksen jälkeen seuraavia sisältötyyppejä: kokoelma, tapahtuma, tiedosto, kansio, kuva, linkki, uutinen ja sivu. Sisältötyypit lisätään sisällönhallintapalkin 'Lisää uusi...' -option alta. Sisältö luodaan sivun sen hetkiseen sijaintiin, joten käyttäjän tulee navigoida sivulla uuden sisällön haluttuun sijaintiin ennen sisällön lisäämistä. Sisältöjä voi jälkikäteen kopioida, leikata ja liittää sijainnista toiseen jo aiemmin mainitulta sisältö-välilehdeltä. (Content Types 2015.)

Tiedostot, kansiot, kuvat ja linkit ovat nimiensä veroisia yksinkertaisia sisältöjä. Tapahtuma on sivun kaltainen sisältö, joka pitää sisällään tapahtuman ajankohdan, sijainnin ja osallistujat. Käyttäjien on mahdollista lisätä tapahtuma eri kalenterisovelluksiin, kuten Google- tai Outlook-kalenteriin. Uutinen muistuttaa rakenteeltaan tapahtumaa, mutta on mukautettu uutisten kirjoittamista varten. Uutisiin voi oletuksena lisätä kuvan, joka näkyy uutisen tiivistelmän yhteydessä. Sivun on Plonen suora-

viivaisin sisältötyyppi. Sen avulla on helppo luoda suurin osa verkkosivun yleisestä sisällöstä. (Mt.)

Kokoelmat mahdollistavat samankaltaisten sisältöjen niputtamisen yhteen paikkaan sivustolla. Esimerkiksi voidaan listata kaikki sivuston kuvat näytettäväksi yhdellä sivulla niiden lisäämisjärjestyksessä. Kokoelma ei siis itsessään sisällä tiedostoja kansioiden tapaan, vaan sen sijaan käyttäjä määrittelee kokoelman ehdoilla mitkä sisällöt näytetään. (Introduction to Collections 2015.)

2.3 Deliverance

Plonen teemoitus toteutetaan nykyään Diazon avulla. Diazo puolestaan syntyi Deliverancen pohjalta. Kaikki alkoi pyrkimyksestä toteuttaa pysyvä ulkoasuratkaisu alustalle kuin alustalle ilman, että kehittäjän pitää tehdä tyylittely aina uudestaan seuraavaan alustaan vaihdettaessa. Näin syntyi Deliverance. Sen tavoitteena oli toimia integrointityökaluna mahdollistaen yhtenäisen tyylittelyn eri alustojen välillä. Deliverancen peruseriaatteena oli toteuttaa HTML-tyylimuunnos käyttäen Pythonia. (Bicking 2012.)

2.4 Diazo

Kolme Deliverancen kehittäjää jatkokehittivät Deliverancen pohjalta ratkaisua, joka sai aluksi nimen XDV, joka loi tyylimuunnosta varten XSLT-tiedoston. Tämä nopeutti tyylimuunnoksen suorittamista. Vuonna 2010 Plone konferenssissa XDV nimettiin Diazoksi epämääräisyyden vähentämiseksi, sillä lyhennettä XDV oli jo käytetty kahdessa eri Plone-komponentissa. Diazosta on muotoutunut De facto standardi Plonen tyylittelemiseen versiosta 4.2 eteenpäin.

Diazo mahdollistaa staattisen verkkosivun tyylin lisäämiseen minkä tahansa serveripuolen dynaamisesti toteutetun verkkosivun käyttöön. Tämä mahdollistaa ulkoisen verkkosuunnittelijan suunnitteleman teeman lisäämisen järjestelmään ilman, että kumpikaan osapuoli edes katsoo toistensa tuottamaa lähdekoodia. Diazoteemat koostuvat kolmesta elementistä: itse ulkoasusta, joka määrittelee sen, miltä sivun tulisi näyttää, sisällöstä, joka on tuotettu Plonen tai jonkun muun järjestelmän puolelle sisällöksi teemoitettavalle sivulle, sekä sääntötiedostosta. Sääntötiedosto määrittelee XML-syntaksia hyödyntäen, mihin sisältö sijoitetaan uudessa teemassa.

Säännöt on määritelty sekä Diazon kotisivuilla että Plonen dokumentaatiossa. Säännöt hyödyntävät CSS3-ilmauksia tai vaihtoehtoisesti XPath-ilmauksia. Molempia saa käyttää vapaasti samaa sääntötiedostoa luodessa, mutta yksittäinen sääntö saa sisältää vain yhdenlaisia ilmauksia. (plone.app.theming manual 2015.)

Alla olevassa rules.xml-tiedostossa ovat näkyvissä tarvittavat viittaukset Diazoon, CSS- ja XML-viittauksiin. Sääntö määrittelee ensimmäisen käytettävän teeman indeksisivun, jonka päälle Plone-sivuston sisältöä sovitetaan. Tämän jälkeen määritellään kaikki main-luokan sisältävien div-elementtien lapset korvattavaksi Plonen määrittämällä div-elementeillä, joiden id-attribuutin arvona on "content". Lopuksi korvataan teeman footer-elementin sisällä oleva teksti Plonen footer-elementiksi määritellyllä div-elementin sisällöllä.

```
<?xml version="1.0" encoding="UTF-8"?>
<rules
  xmlns="http://namespaces.plone.org/diazo"
  xmlns:css="http://namespaces.plone.org/diazo/css"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

  <theme href="index.html" />
  <replace css:content="#content"
    css:theme-children=".main" />
  <replace css:content="#portal-footer p"
    css:theme="footer p" />
</rules>
```

Kuviossa 3 on näkyvissä sivun sisältö Plonen vakioulkoasu vasemmalla sekä yllä luodun rules.xml-tiedoston kanssa Diazo-teemalla oikealla.



Kuvio 3. Plonen tavallinen ulkoasu vasemmalla, Diazo-teema oikealla

3 Responsiivinen web-suunnittelu

3.1 Lähtökohta

Vuonna 2010 Ethan Marcotte esitteli verkkosivujen kehittämisen ongelmakohtia artikkelissaan Responsiivinen verkkosuunnittelu. Tuolloin käytäntönä oli vielä luoda erilliset verkkosivut mobiilikäyttöön, mutta ongelmaksi muodostuivat laitteiden jatkuvasti kasvava kirjo sekä käyttäjämäärät. (Marcotte 2010.)

Ensimmäisenä ratkaisuna Marcotte ehdotti verkkosivua, joka vain yksinkertaisesti skaalaa sisällön resoluution mukaan, ja esitteli samalla siitä aiheutuvat ongelmat. Tekstin luettavuus katoaa, kuvista tulee liian pieniä ja ne menettävät yksityiskohtansa tai jopa joudutaan rajaamaan tiukasti niin, että osa kuvaa katoaa täysin. Myös suuret resoluutiot ja laajakuvanäytöt tuovat mukanaan ongelmia, kuvista tulee liian isoja ja ne valtaavat huomion muulta sisällöltä. (Mt.)

Responsiivisen web-suunnittelun pohjimmainen tarkoitus on MarcOTTen mukaan luoda yhtenäinen käyttökokemus päätelaitteesta riippumatta. Sen sijaan että yritetään luoda jokaiselle laitteelle oma käyttöliittymä, kehittäjien tulisi keskittyä luomaan yksi yhtenäinen mukautuva käyttökokemus. On helpompaa keskittyä ulkoasun murtumispisteisiin kuin tiettyjen päätelaitteiden näyttöjen kokoon. Ethan Marcotte to-

teutti tämän ajatusmallin käyttäen joustavaa ruudukkoa, kuvia sekä mediakyselyjä.
(Mt.)

3.2 Mukautuva ruudukko

Vanhempi vakinaistunut käytäntö oli sitoa sivun sisältö tietyn levyisiin pystyriveihin eli ruudukkoon (grids). Näiden pystyrivien mitat määritettiin pikseleissä, esimerkiksi seitsemän 144 pikseliä leveitä pystyrivejä 20 pikselin välityksellä koostaen sivun leveydeksi 988 pikseliä. Käyttäjiltä oletettiin tiettyä resoluutiota, eli toisin sanottuna luvattiin tukea kaikkia käyttäjiä, joiden vaakaresoluutio ylitti pyydetyn minimin. Tämä osoittautui tehokkaaksi ja näppäräksi tavaksi toteuttaa verkkoulkoasuja työpöytäkäyttöön. (Marcotte 2009.)

Ratkaisu pienempien vaakaresoluutioiden tukemiseen löytyi Ethan Marcotten tapauksessa Internet Explorerin kyvyttömyydestä tukea pikselikoossa määritellyjä fontteja. Fonttikoon sai kuitenkin asetettua haluamakseen käyttämällä suhteellisia yksiköjä, kuten avainsanoja, prosenttikokoja tai tässä tapauksessa em-yksiköjä. Em-yksikkö on sidottu elementin "äitielementin" fonttikokoon. Kun fonttikoko on asetettu sivulla sataan prosenttiin, se useimmissa tilanteissa tarkoittaa pikselikokoa 16. Mikäli haluttaisiin asettaa esimerkiksi otsikon fontti pikselikokoon 24, sen voi tehdä käyttämällä em-yksiköjä laskemalla seuraavan yhtälön:

$$\text{kohde / konteksti} = \text{tulos}$$

Sijoittamalla aiemmin mainitut luvut saadaan

$$24 / 16 = 1,5$$

Näin saadaan koko em-yksikköinä. Tästä Ethan Marcotte sai idean sitoa pystyrivien leveys suhteellisiin yksiköihin. Esimerkiksi sivun leveys, joka oli pikselikokoon sidottuna 988 pikseliä leveä fontin peruskokoon ollessa 16 pikseliä, sen voi määrittää suhteelliseksi äsken esitellyllä kaavalla:

$$988 / 16 = 61,75$$

Täten sivun leveydeksi voidaan asettaa 61,75 em. Seuraavana voitaisiin esimerkiksi asettaa sivun otsikko, joka oli 700 pikseliä leveä.

$$700/988 = 0,7085$$

700 pikseliä on siis 70,85 % sivun maksimileveydestä. Samaa periaatetta noudattamalla saadaan sivun muukin tekstisisältö sidottua suhteellisiin yksikköihin. (Mt.)

3.3 Mukautuvat kuvat

Kuvien kohdalla Ethan Marcotte noudatti samoja periaatteita kuin joustavan ruudun kanssa. Asettamalla kuvan kehyksen haluttuun suhteelliseen kokoon tulee vielä määrittää kuvaelementin maksimileveydeksi 100 %. Näin menetellessä kuva ei pääse "vuotamaan" kehyksen ulkopuolelle, vaan se skaalautuu kehyksen mukaan. (Marcotte 2011.)

3.4 Mediakyselyt

Mediakyselyt (media queries) mahdollistavat tyylien asettamisen verkkosivuille laitteen toistoresoluutioiden mukaan. Sivuille voidaan esimerkiksi määrittää minimikapeus, jonka jälkeen vaihdetaan kokonaan toiseen tyyli tiedostoon. Alla olevassa esimerkissä vaihdetaan lowres.css-tyylitiedostoon, mikäli päätelaitteen maksimi ruudunleveys on 480 pikseliä tai sen alle.

```
<link rel="stylesheet" type="text/css"
media="screen and (max-device-width: 480px)"
href="lowres.css" />
```

Tarvittaessa on myös mahdollista ketjuttaa useampi and-ehto, jolloin ehto toteutuu vain, jos molemmat määritelmät ovat voimassa:

```
<link rel="stylesheet" type="text/css"
media="screen and (max-device-width: 480px) and (resolution:
162dpi)" href="lowres.css" />
```

Lisäksi kyselyitä voi hyödyntää myös tyyli tiedoston sisällä:

```
@media screen and (max-device-width: 480px) {
  .column {
    float: none;
  }
}
@import url("lowres.css") screen
and (max-device-width: 480px);
```

Jokaisen esimerkin kohdalla tavoite kuitenkin pysyy samana, mediakyselyillä asetetaan tyyllittelylle ehtoja, joiden avulla voidaan tarkastella päätelaitteen ominaisuuksia ja mukauttaa sivua sen mukaan. On mahdollista esimerkiksi kasata kaikki sivun ele-

mentit päällekkäin ja ryhmitellä kuvat yksitellen allekkain, kun katselijan vaaka-resoluutio alkaa tuottaa ongelmia normaalin ulkoasun kanssa. Kuten alla olevasta kuviosta 4 ilmenee, mediakyselyitä on tietysti myös mahdollista käyttää toisinkin päin, jos vaakaresoluutio riittää vaikka kuuden kuvan rinnakkain aseteltuun kolmen sijaan, sekin onnistuu. (Marcotte 2010.)

Otsikko

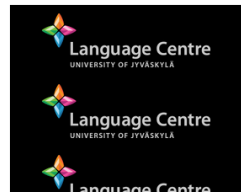
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Commodi, quaerat, mollitia, odit vitae esse obcaecati voluptatibus pariatur atque quis assumenda eius deserunt corporis ab a consectetur ex officis dolorum sunt? Lorem ipsum dolor sit amet, consectetur adipiscing elit. Commodi, quaerat, mollitia, odit vitae esse obcaecati voluptatibus pariatur atque quis assumenda eius deserunt corporis ab a consectetur ex officis dolorum sunt? Lorem ipsum dolor sit amet, consectetur adipiscing elit. Commodi, quaerat, mollitia, odit vitae esse obcaecati voluptatibus pariatur atque quis assumenda eius deserunt corporis ab a consectetur ex officis dolorum sunt?



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Commodi, quaerat, mollitia, odit vitae esse obcaecati voluptatibus pariatur atque quis assumenda eius deserunt corporis ab a consectetur ex officis dolorum sunt?

Otsikko

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Commodi, quaerat, mollitia, odit vitae esse obcaecati voluptatibus pariatur atque quis assumenda eius deserunt corporis ab a consectetur ex officis dolorum sunt?



Kuvio 4. Esimerkki kuvien asettelusta eri kuvasuhteita tukeville näytöille

3.5 Nykyinen tilanne

Ethan Marcotten luomista määritelmistä on kulunut viisi vuotta, mutta yhä ne pitävät paikkansa, suurimmaksi osaksi. Artikkelissa "The State of Responsive Web Design" (Bracey 2015) määriteltiin responsiivisuus seuraavasti:

Responsiivisen web designin tavoite on luoda verkkosivuja joiden sisältö ja käytettävyys on optimoitu parhaalla mahdollisella tavalla selaimesta tai päätelaitteesta riippumatta. (Bracey 2015.)

3.6 Vakiintuneet käytännöt

Nykyisten laitteiden resoluutiokirjo on uskomattoman laaja. Myös kosketus- ja ele-tunnistus on yleistynyt ja otettava huomioon. On turhaa yrittää tukea jokaista laitetta erikseen. Sen sijaan on hyvä keskittyä tunnistamaan ulkoasun hajoamispisteet ja muokata ulkoasu kuntoon niiden kautta. Kuvien tulee olla joustavan kokoisia, että ne aina näyttävät hyvältä laitteesta huolimatta. Myös muun median, kuten video- ja audiosoitinien, tulee käyttäytyä samoin. Mobiililaitteet on asetettu olettaamaan, että niiden tulisi näyttää koko sivu, mikäli niitä ei erikseen ohjeisteta. Siksi on syytä käyttää viewport-metatagia, jolla voi ilmoittaa päätelaitteelle, että tuki pienemmille tarkkuuksille löytyy. (Bracey 2015.)

3.7 Vanhentuvat käytännöt

Laitteiden kirjo on liian suuri, että olisi turvallisesti mahdollista erottaa mobiililaitteet työpöytälaitteista. Esimerkiksi kun isoa resoluutiota tukevat kannettavat laitteet omaavat kosketustuen samaan aikaan kun pieniresoluutioinen netbook ei, on äärimmäisen vaikea määrittellä, millä resoluutiolla tulisi mahdollistaa kosketustuki.

Kiinteän pikselikoon ulkoasuista on luovuttu jo aikaa sitten, ja se on tapahtumassa myös lopuille elementeille. Em-, rem- ja prosenttiyksiköt mahdollistavat tasaisen skaalauksen resoluutiosta huolimatta.

Flash-sisällön rooli on jatkuvasti pienentynyt, mutta siihen pohjautuvia videoita sekä pelejä näkee yhä ajoittain. Ongelmaksi nousevat turvallisuus sekä vakaus. HTML5 mahdollistaa tuen vastaavalle sisällölle ilman selaimen lisäosia, joten se on turvallisempi valinta. (Bracey 2015.)

3.8 Uudet tulokkaat

Natiivit HTML5-toteutukset valtaavat alaa vanhemmilta selainliitännäisiltä kuten Flashilta. HTML5-video ja audiosoittimet, uusi HTML5:n picture-elementti sekä työkalut kuten Canvas ja WebGL mahdollistavat monenlaista sisältöä. Palvelut kuten Font awesome, Entypo ja Typicons mahdollistavat kaikkennäköisiä ikoneja sivuille fontteina toteutettuna. Tämä tarkoittaa sitä, että niiden väriä ja kokoa voi muokata suoraan CSS-koodin kautta parantaen skaalautuvuutta. (Bracey 2015.)

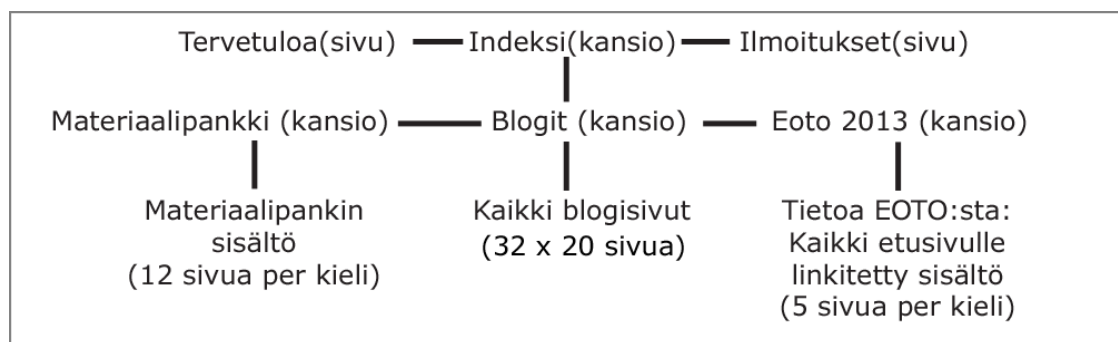
4 Each One Teach One -sivuston ulkoasu-uudistus

4.1 Mikä on EOTO?

EOTO on kielikeskuksen ohjaama pari- tai ryhmätyöskentelyohjelma, jossa opiskelijat opettavat toisilleen omaa äidinkieltään tai koulukieltään ja opiskelevat parinsa kieltä. Ohjelman sivustolla kaikki materiaali löytyy suomeksi ja englanniksi. Koska tässä oppinäytetyössä keskityttiin uudistamaan ulkoasu opiskelijoita silmällä pitäen, sivuston toimintaa ei tarkasteltu hallinnoijan eikä opettajan näkökulmasta.

Sivusto tukee ohjelman suorittamista opiskelijoiden näkökulmasta tarjoamalla heille mahdollisuuden kirjautua sivustolle. Tunnukset tulee pyytää ensin yhdeltä ohjelman

ohjaajista. Tunnusten luonnin jälkeen Opiskelijat voivat joko luoda itsestään ilmoituksen parin etsintää varten Ilmoitukset-sivulle tai etsiä itseään kiinnostavan parin jo luoduista ilmoituksista. Kun pari tai ryhmä on saatu kasaan, pyydetään oikeus omaan blogiin. Blogisivulla on viisi ohjelman suoritukseen sidottua lomaketta sekä tilaa blogimerkinnöille. Lomakkeista täytetään kaksi ensimmäistä ja tämän jälkeen sovitaan tapaaminen ohjaajan kanssa. Tämän jälkeen opiskelijat sopivat tapaamisista ja täyttävät blogi-sivulle merkinnän jokaisesta tapaamisesta, joita on vähintään kahdenkymmenen tunnin edestä. Loput lomakkeet täytetään suorituksen edetessä. Sivusto tarjoaa myös lisäinformaatiota kurssin suorittamista varten Materiaalipankki-sivun muodossa. Sivuston rakenteen näkee paremmin alla olevasta kuvioista 5. Se koostuu neljästä kansioista ja useasta erillisestä sivusta.



Kuvio 5. Vanhan sivuston kansiorakenne

4.2 Vanha ulkoasu

Etusivu koostui pääosin tekstisisällöstä. Sivun vaikutti pieneltä nykyaikaisten työpöytänäyttöjen resoluutioilla, eikä mobiililaitteita ollut otettu huomioon. Navigaatio löytyi sivun oikeasta laidasta sisäänkirjautumis- ja kielenvaihto-option kera. (Ks. kuvio 6.) Kuviossa 5 mainitut tietoa EOTO:sta -sivujen sisältö oli olennaista uusille osallistujille, mutta ne oli ainoastaan linkitetty etusivun tekstin sekaan. Sivut lisätiin uudessa ulkoasussa suoraan päänavigaatioon. Suurin osa palvelun rakenteesta koostui vastaavanlaisista sivuista. Mukaan mahtui myös muutama poikkeus.

EOTO
EACH ONE TEACH ONE

Tervetuloa Each One Teach One -sivuille!

Each One Teach One -ohjelma eli EOTO on kolmen opintopisteen laajuinen ohjelma sinulle, joka haluat tutustua erimaalaisiin opiskelijoihin ja oppia vieraita kieltä. Ohjelmassa opiskelijat opettavat toisilleen omaa äidinkieltään tai koulukieltään ja opiskelevat parinsa kieltä. Parityöskentelyn lisäksi kieliä voidaan opiskella myös pienissä ryhmissä, ja opiskelijat voivat perustaa myös yhteen kieleen keskittyviä keskusteluryhmiä ja lukupiirejä.

Kiinnostuitko? Jatka siis eteenpäin ja tutustu tarkemmin Each One Teach One -ohjelmaant!

[Mikä ihmeen EOTO?](#) -sivulla kerrotaan EOTO-ohjelmasta ja sen suorittamisesta.

[Ilmoitukset](#)-sivun kautta voit löytää itsellesi parin! Lue muiden opiskelijoiden hakuilmoituksia tai jätä oma ilmoituksesi.

- Etusivu
- Ilmoitukset
- Materiaalipankki
- Blog

[Kirjaudu](#)

Yhteystiedot/Contact info:
ecoteachers@ulc.jyu.fi

Käyttäjätunnus

Salasana

Kirjaudu sisään

Salasana hukassa?

English

Suomi

Jyväskylän yliopiston kielikeskus, PL 35 (Oppio), 40014 Jyväskylän yliopisto. (c) Kielikeskus, kaikkinaisen lainaaminen ilman kielikeskuksen lupaa kielletty.

Kuvio 6. Vanhan ulkoasun etusivu

Sivuston Ilmoitukset koostuivat taulukosta, jonka tarkoituksena oli listata kurssilais-ten ryhmän tai työparin hakuilmoitukset. Taulukon pystyi järjestämään minkä tahansa sarakkeen mukaan, mikäli käyttäjä oli kirjautunut sisään. Vaakaresoluution kave- tessa taulukolta loppui tila nopeasti. (Ks. kuvio 7.)

Ilmoitukset

Koko nimi	Sukupuoli	Äidinkieli	Kieli jota haluaa opiskella	Lisätiedot
[REDACTED]	Male	Catalan and Spanish	Finnish and English	Näytä koko ilmoitus
[REDACTED]	Female	Ukrainian, Russian	Finnish	Näytä koko ilmoitus
[REDACTED]	Male	Spanish	Finnish	Näytä koko ilmoitus
[REDACTED]	Female	Russian	German, Portuguese, French	Näytä koko ilmoitus
[REDACTED]	Female	Spanish	English	Näytä koko ilmoitus
[REDACTED]	Female	Finnish, French, Arabic	Spanish	Näytä koko ilmoitus
[REDACTED]	Female	Czech	English	Näytä koko ilmoitus
[REDACTED]	Female	Finnish	Spanish	Näytä koko ilmoitus

[Lisää oma ilmoitus](#)

- Etusivu
- Ilmoitukset
- Materiaalipankki
- Blog

[Kirjaudu](#)

Yhteystiedot/Contact info:
ecoteachers@ulc.jyu.fi

Käyttäjätunnus

Salasana

Kirjaudu sisään

Salasana hukassa?

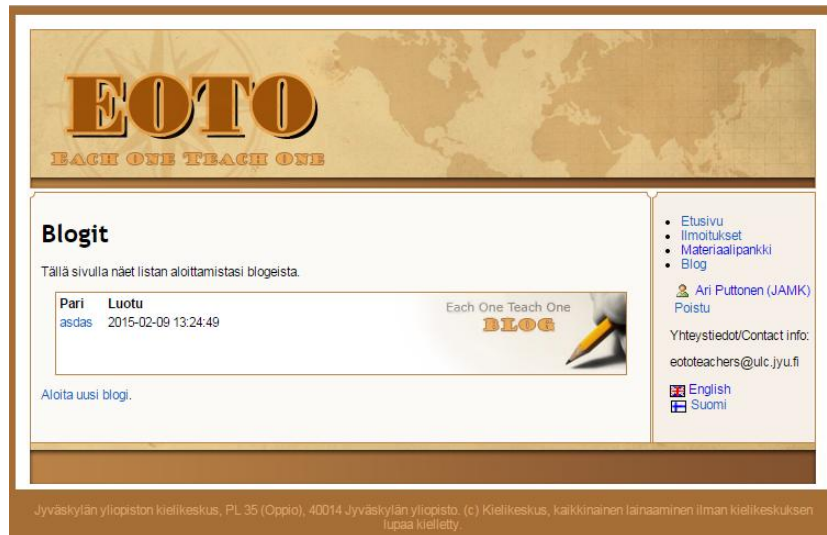
English

Suomi

Jyväskylän yliopiston kielikeskus, PL 35 (Oppio), 40014 Jyväskylän yliopisto. (c) Kielikeskus, kaikkinaisen lainaaminen ilman kielikeskuksen lupaa kielletty.

Kuvio 7. Vanhan ulkoasun Ilmoitukset -sivu

Sivuston Blogit-sivu listasi tyyliteltyyn taulukkoon käyttäjän aloittamat blogit. Bloggeja pääsi luomaan vain kirjautuneena käyttäjänä. Opettajat pääsivät käsiksi kaikkiin blogeihin tämän sivun kautta. (Ks. kuvio 8.)



Kuvio 8. Vanhan ulkoasun Blogin etusivu


Blogit koostuivat blogaajan tiedoista listattuna sivun ylälaitaan, viidestä kurssin aikana täytettävästä lomakkeesta sekä jokaisen tapaamisen jälkeen luotavista blogimerkinnöistä. (Ks. kuvio 9.) Blogimerkinnöille sivun vasemmassa laidassa jäi kapea tila lomakelinkkien ollessa niiden oikealla puolella. Merkintöjä tulee kurssin toteuttamisen aikana yli kaksikymmentä, joten lista tulisi jatkumaan paljon esimerkkikuvaa pidemmälle.



Kuvio 9. Vanhan ulkoasun blogisivun sisältösivu

4.3 Uusi ulkoasu

Sekä toimeksiantaja että asiakas antoivat vapauden toteuttaa uuden ulkoasun sivustolle ilman ennakkovaatimuksia. Kun ensimmäinen ehdotelma oli valmis, se käytiin läpi palaverissa, jossa kaikki osapuolet olivat läsnä. Ehdotus miellytti asiakasta ja toimeksiantajaa. Verkkosivupohjan kehittäminen aloitettiin tapaamisen jälkeen konseptikuvan pohjalta. Alla olevasta kuvista 10 näkee ulkoasun ennen kuin sitä alettiin muuttaa Ploneteemaksi Diazon avulla. Ulkoasussa hankkiuduttiin eroon sivun oikeassa laidassa olevasta navigaatiosta ja siirrettiin se sisällön yläpuolelle. Näin se oli helpommin havaittavissa antaen samalla lisää tilaa sisällölle. Vanha ulkoasu oli 745 pikseliä leveä riippumatta toistolaitteesta. Uusi ulkoasu oli maksimileveydessään vasta 1056 pikselin kohdalla kaveten tarvittaessa. Myös sisältötekstin kokoa kasvatettiin luettavuuden parantamiseksi kuudentoista pikselin korkuiseksi.

EOTO  [LOG IN](#)

Each One Teach One

[Home](#) [Information](#) [Advertisements](#) [Ideas & Materials](#) [Blog](#)

Information about EOTO

The basic idea

Each One Teach One is a pair/group programme coordinated by the University of Jyväskylä Language Centre. The students teach each other their mother tongues or the languages of their education, and study the language of their partner. The students should be native-like in the language they teach, but otherwise the students aren't expected to have had any training as a teacher or language teacher.

Studying

You meet with your partner regularly, usually once or twice a week. Each language should be studied for at least 20 hours face to face (20+20 hours). In addition to the meetings, you complete homework exercises and write about each meeting on a personal blog which the EOTO teachers follow. The blog also contains forms to fill in to analyse the learning process.

Don't ask your partner to look for exercises or produce learning material! Each student is responsible for their own material and learning.

You receive 3 ECTS credits for the programme. The EOTO credits count towards the elective studies of the language that the student has been studying. You can participate in EOTO as many times as you want, but you cannot use it to replace other Language Centre courses. You can start the programme at any point of the semester, but it should be completed within the same academic year.

The phases of the programme

EOTO has three phases: planning, active study and wrap-up. Together they form the EOTO path. The purpose of the EOTO path is to encourage and support your studies and give new ideas for language learning.

Kuvio 10. Uuden ulkoasudemon informaatio sivu

4.4 Tyhjän pohjan luominen

Jotta ulkoasusta saadaan toimiva teema, se täytyy ensin muuttaa tavalliseksi pelkistetyksi verkkosivuksi, joka taas voidaan myöhemmin muuttaa teemaksi. Prosessi aloitettiin kirjoittamalla etusivulle sisällötön HTML-pohja hyödyntäen HTML5 Boilerplatea ja Normalizer.css-tiedostoa, joka yhdenmukaistaa eri selainten oletustyyli määrittelyjä. Lisäksi tarvittiin jQuery JavaScript-kirjastoa eri selainten yhteensopivuuden takaamiseksi.

HTML5 Boilerplate on verkkoapplikaatioita tai sivuja varten luotu pohja, jonka tarkoituksena on nopeuttaa halutun projektin aloittamista. Boilerplatea käytettiin EOTO-sivun teeman luonnin aloittamisen apuna. Se pitää sisällään väliaikaisiesimerkin kosketusnäyttölaitteille suunnatusta ikonista, Modernizr-kirjaston, jonka tarkoitus on tarkistaa selaimien tukemat verkkotekniikat sekä optimoidun Google Analytics -skriptin, joka mahdollistaa palvelun kävijöiden analysoinnin. Lisäksi mukana on jQuery Javascript-kirjasto Googlen hostauspalvelun välittämänä sekä Normalize.css-tiedosto.

Normalize.css on pieni CSS-tiedosto, joka asettaa verkkosivulle yhtenäisemmän ulkoasun vähentäen eri selainten aiheuttamia eroavaisuuksia verkkosivun tulkinnassa.

Sen tavoitteena on pitää hyödylliset selainkohtaiset asetukset, mutta samalla normalisoida tyylit, korjata virheet ja parantaa käytettävyyttä. Normalize.css ei siis ole tyylinollaustyökalu, vaan sitä voi käyttää verkkosivun ulkoasun tyylipohjana muokkaamalla tiedostoa suoraan oman maun mukaan, tai vaihtoehtoisesti ajamalla sen yli omalla erillisellä tyylitiedostolla. (Callagher 2012)

jQuery on nopea, pieni ja ominaisuuksiltaan rikas JavaScript-kirjasto. Sen valtteja ovat näppärä dokumenttiolionmallin hyödyntäminen, tapahtumien hallinnan helpottaminen, Ajax-kutsujen yksinkertaistaminen sekä erityisesti yhteensopivuus useiden selainten kanssa. (jQuery 2015.)

Koska lähes kaikki EOTO-sivuston sisältö tulee Plone-elementtien kautta, niiden toiminnallisuus jätettiin Plonen harteille. Täten ei ollut tarvetta käyttää Modernizr-kirjastoa eri selainten tuettujen elementtien tutkimiseen. Viittaukset kyseiseen kirjastoon voitiin pyyhkiä teeman indeksisivulta. Sivuston sisältämä kieli ilmoitettiin HTML-elementin sisällä. Sivuston merkistö tulee määrittää tiedoston ensimmäisten 1024 tavun aikana ja mahdollisimman pian haavoittuvuuksien välttämiseksi. Täten se asetettiin ennen sivuston title-elementtiä. Internet Exploreria käyttäessä on mahdollista, että käyttäjä on asettanut selaimensa käyttämään vanhentunutta selainrajapintaa. Tällöin sivusto voitiin asettaa käyttämään viimeisintä tarjolla olevaa rajapintaa meta-elementillä, jossa content-attribuuttin arvoksi asetettiin ie-edge. Sivuston viewportin, eli näkymän sopiva skaalaus asetettiin eri laitteita varten täällä. Jos skaalusta ei aseteta, eri laitteet, kuten vaikka älypuhelin, olettaisi että sen olisi tarkoitus näyttää koko sivu simuloiden isompien monitorien käytöstä. Samalla kaikki responsiivisuutta varten kirjoitettu koodi jäisi hyödyntämättä. Seuraavaksi linkitettiin sivuston suosikki- ja kosketusikonit. Viimeisenä asetettiin linkki normalize.css-tiedostoon ja sivuston tyylitiedostoon. Näiden päätösten seurauksena teemapohjan HTML-tiedosto näytti tältä:

```
<!doctype html>
<html lang="fi">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <title>EOTO</title>
    <meta name="description" content="">
    <meta name="viewport" content="width=device-width,
    initial-scale=1">
    <link rel="apple-touch-icon"
```

```

        href="apple-touch-icon.png">
        <link rel="shortcut icon" type="image/x-icon"
        href="favicon.ico">
        <link rel="stylesheet" href="css/normalize.css">
        <link rel="stylesheet" href="css/style.css">
    </head>
    <body>
    </body>
</html>

```

Muita HTML5 Boilerplatesta löytyviä määritteitä olivat selaimen päivitysnotifikaatio vanhaa Internet Explorer selainta käytettäessä, linkit jQuery JavaScript-kirjastoon, sivuston muihin JavaScript-tiedostoihin, sekä siistitty Google Analytics -skripti. Päivitysnotifikaatio oli hyödyksi teemassa. Modernizr-kirjasto sivuutettiin Plonen takia, kuten edellä mainittiin. JQueryyn avulla oli mahdollista toteuttaa loput JavaScript-pohjaiset ominaisuudet, joten viitaukset kirjastoon säilytettiin. Sen ansiosta erillisiä JavaScript tiedostoja ei tarvittu. Google Analytics on erinomainen käyttäjästatistiikkapalvelu, mutta teemademonstraatioon sitä ei tarvittu. Myös se voitiin pyyhkiä teemapohjasta ainakin kehittämisen ajaksi. HTML-pohjan body-elementti ilman teeman sisältöä näytti tältä:

```

<body>
  <!--[if lt IE 9]>
    <p class="browserupgrade">You are using an
    <strong>outdated</strong> browser. Please
    <a href="http://browsehappy.com/">
    upgrade your browser</a>
    to improve your experience.</p>
  <![endif]-->
  <script
  src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.
  min.js"></script>
</body>

```

4.5 Esimerkkejä teeman rakenteesta ja responsiivisuudesta

Sisällön rakentaminen alkoi rakenteen hahmottamisesta. Sivun koostui yläotsikosta, sen alapuolelle sijoitetusta sivustonavigaatiosta ja sisältösolusta. Niiden alle asetettiin sivun alatunniste. Sisältö ennen alatunnistetta niputettiin div-elementin sisään, jolle annettiin luokaksi wrap. Alatunniste asetettiin sen alle. div-elementtiä, jonka luokka on container, käytettiin sisällön keskittämiseen. Alla olevassa koodissa on body-elementin sisältö ennen JavaScript-viittausta.


```

<div class="wrap">
</div>
<footer>
  <div class="container">
    <p>Alatunnisteen sisältö</p>
  </div>
</footer>

```

Tyylimäärittelyt elementille footer, sekä luokille wrap ja container, ovat nähtävissä alapuolella olevassa CSS-tyylimäärittelyssä. Wrap-luokka asetettiin koko selaimessa olevan tilan korkuiseksi. Tällöin footer-elementille asetettu negatiivinen ylämarginaali ja samansuuruinen mutta positiivinen korkeus lukitsivat elementin sivun alalaitaan. Tämä estää footer-elementin nousemisen ylemmäs, mikäli sivulla ei ole tarpeeksi sisältöä. Samalla varmistettiin elementin leveys koko sivun levyiseksi puhdistamalla sisältö sen molemmilta puolilta. Container-luokalla asetettiin sivun sisällön maksimileveys ja samalla pakotettiin se kaventumaan, mikäli koko sivun leveys putoaa elementissä määritellyn maksimileveyden alle.

```

footer {
  position: relative;
  margin-top: -12em;
  height: 12em;
  clear: both;
  background-color: #373737;
}

.wrap {
  min-height: 100%;
}

.container {
  max-width: 59.375em;
  width: 90%;
  margin: 0 auto;
}

```

Mobiilinnavigaatiota varten tarvittiin JavaScriptiä. Joustavuuden ja laajan yhteensopi-
vuuden kannalta päädyttiin käyttämään kappaleessa jo aiemmin mainittua jQuery
JavaScript-kirjastoa. Mobiilinnavigaation ideana on luoda vaihtoehtoinen kapeampaan
tilaan mahtuva navigaatio. Kun sivuston leveys tippuu mediakyselyn avulla määritel-
lyn rajan alle, piilotetaan normaali inline-block-elementeillä luotu navigaatio ja luo-
daan tilalle navigaatiopainike.

```

@media screen and (max-width: 678px) {
  .menu { display: none; }
  .menu.open { display: block; }
}

```

```
.mobile-navi-button { display: block; }
.menu-item { display: block; }
}
```

Navigaatiopainike luo uuden open-luokan piilotetulle navigaatiolle jQueryn avulla.

```
<script>
  $(' .mobile-navi-button').bind('click', function() {
    $(' .menu').toggleClass('open');
  });
</script>
```

Uusi luokka tuo tavallisen navigaation näkyviin, mutta elementit on muutettu block-tyyppiin, jolloin ne järjestyvät allekkain.

Ideat & materiaalit -sivu sisälsi ainoastaan linkkejä toisille yksinkertaista tekstisisältöä sisältäville sivuille. jQueryn avulla oli mahdollista tuoda alasivujen sisältö suoraan Ideat & materiaalit -sivulle. Tavallisten linkkien sijaan klikattavat otsikot tuovat alasivujen sisällön suoraan näkyviin otsikkojen alle. JQuery lisää piilotetulle sisällölle luokan otsikkoa klikatessa tuoden sisällön näkyviin. Ainut ero verrattaessa mobiilinnavigaatioon on se, että luokka lisätään seuraavalle tagisolmulle.

```
$('.expandable-title').bind('click', function() {
  $(this).next().toggleClass('open');
});
```

Responsiivisuuteen törmättiin myös Ilmoitukset-sivun kohdalla, joka sisälsi taulukon. Taulukossa oli viisi leveää saraketta, joiden ansiosta sivun vaakaleveyden kavetessa tila loppui nopeasti kesken. Vastaus ongelmaan löytyi mediakyselyistä. Kun sivuston leveys putoaa alle taulukon leveyden, piilotettiin taulukon otsikkorivi ja listattiin kaikki yhden rivin sarakkeet yhteen sarakkeeseen allekkain. Tämä toteutettiin alla olevassa koodissa. Sivun leveyden pudotessa alle 860 pikselin taulukon otsikko piilotettiin, sen sisältösolut pakotettiin allekkain asettamalla niiden esitysmuoto block-muotoon ja ensimmäisen sisältösolun teksti lihavoitiin. Taulukon rivit esitettiin inline-block-muodossa, jotta voitiin esittää kahden rivin sisältö rinnakkain, kun samalla asetettiin niiden leveys viiteenkymmeneen prosenttiin.

```
@media screen and (max-width: 860px) {
  thead {
    display: none;
  }
  td {
```

```

        display: block;
    }
    td:first-child {
        font-weight: bold;
    }
    tr {
        border-top: 1px solid #ddd;
        display: inline-block;
        width: 50%;
    }
}

```

Kahden rivin sisältö saatiin mahtumaan kahteen rinnakkaiseen sarakkeeseen. Kun vaakaleveys kapenee entisestään, näytettiin sarakkeet allekkain yksi per rivi. Alla oleva koodi asetti yhden rivin sisällön leveyden sataan prosenttiin sivun leveyden kavetessa alle 480 pikselin.

```

@media screen and (max-width: 480px) {
    tr {
        width: 100%;
    }
}

```

Kun float-arvoja käytetään div-elementtien sisällä, ne saattavat päästä "vuotamaan" halutun div-elementin ulkopuolelle. Ongelmaan on ratkaisu, jota kutsutaan nimellä clearfix. Sen ideana on luoda CSS-tyylimäärityillä näkymätön koko sivun levyinen kaistale halutun div-elementin alapuolelle. Alla oleva koodi näyttää, miltä clearfix-luokka näytti CSS-tiedostossa. Ominaisuus päätettiin antaa myös teemassa esiintyvälle container-luokalle luokkien liiallisen käytön vähentämiseksi.

```

.container:after, .clearfix:after {
    display: block;
    content: "";
    clear: both;
}

```

Haastavimmaksi sivuksi muodostui Blogisivu. Sivun rakenne poikkesi muista sivuista lisäten linkit kurssin aikana täytettäville lomakkeille sivun sisällön rinnalle kaventaen sen tilaa entisestään. Ongelma ratkaistiin tuomalla linkit blogin pitäjän tietojen rinnalle sivun yläosaan. Näin itse blogi eli sivun varsinainen sisältö sai koko palstan sisällön itselleen. Sivun kavetessa linkit lomakkeisiin siirretään käyttäjän tietojen alle jo tutuksi tulleilla mediakyselyillä. (Ks. kuvio 11.)



Kuvio 11. Demon Blogisivu täysikokoisena vasemmalla, kapeana oikealla

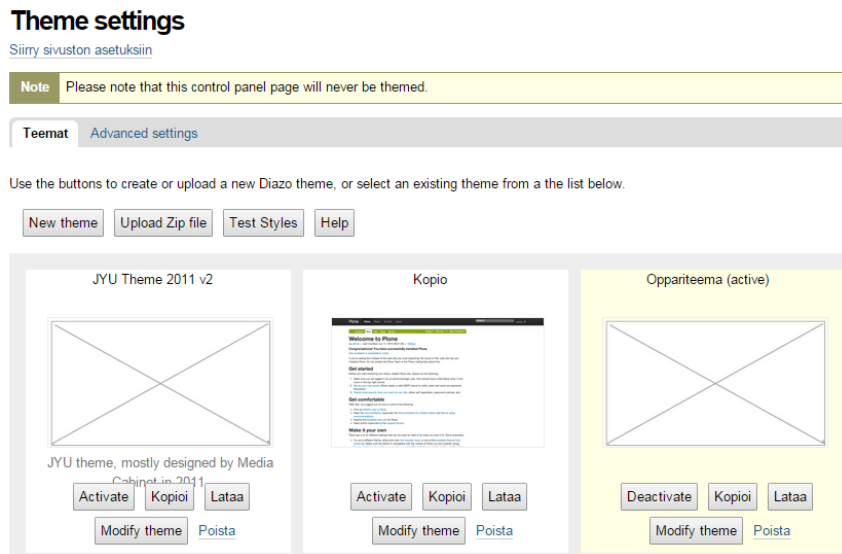
Ulkoasuun päätettiin asettaa kustomoitu fontti. Sen toteuttamiseen käytettiin palvelua nimeltä Google Fonts. Palvelu tarjoaa nopean ja helpon tavan käyttää verkosta saatavia fontteja. Kaikki listatut fontit ovat avointa lähdekoodia, joten niitä saa käyttää ja jakaa eteenpäin vapaasti. Fontin lisääminen sivustolle oli yksinkertaista. Se voidaan lisätä palvelun tarjoamalla viittauksilla suoraan HTML-sivulle, importoida CSS-tiedostossa tai asettaa käyttöön JavaScriptin avulla. Tämän työn kohdalla fontti päätettiin lisätä CSS-tiedostoon, sillä se koettiin osana ulkoasun tyylittelyä. Importoinnin jälkeen fonttia voitiin käyttää tavalliseen tapaan CSS-tiedostosta käsin. Tarvittavat viittaukset ovat nähtävissä alla.

```
@import url(http://fonts.googleapis.com/css?family=Droid+Sans);
body { font-family: 'Droid Sans', sans-serif; }
```

4.6 Plonen teemoitustyökalu

Koska lopulta kaikki sisältö tulee Plonen sisältä, ylimääräinen sisältö karsittiin teemasta. Jäljelle jäi indeksisivu, joka piti sisällään yläotsikon, väliaikaisen linkkipalkin sekä div-elementin sisältöä varten. Lisäksi mukaan tulivat tyylitiedostot ja kuvat sekä logot. Edellä mainitut tiedostot pakattiin ZIP-tiedostoksi ja lähetettiin Plonen tee-

moihin sivuston asetuksissa olevan teemoitus-ominaisuuden avulla. (Ks kuvio 12.)

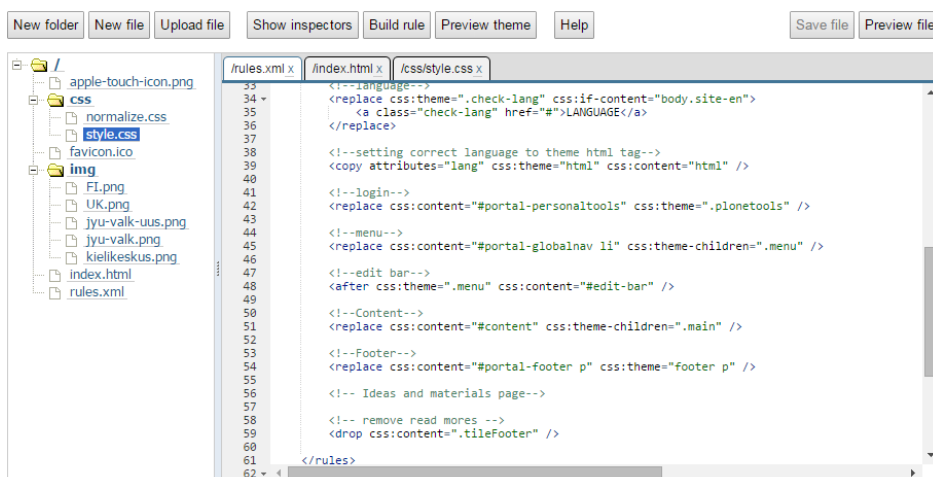


Kuvio 12. Plonen teemoitustyökalu

Teemaa kehitettiin jatkossa Plonen teemoitustyökalun tarjoaman editorin kautta. Editori listasi kaikki teeman tiedostot kansiorakenteineen samalla mahdollistaen niiden muokkaamisen. Tämä koettiin mielekkäämmäksi tavaksi muokata teemaa. Toinen vaihtoehto olisi ollut muokata tiedostoja ulkoisella editorilla, pakata muokattu teema aina uudestaan ja sen jälkeen lähettää teema Ploneen esikatseltavaksi. Näin teeman muokkaamisen ohessa pääsi nopeammin todentamaan tehdyt muutokset suoraan kehittämistä varten pystytetyssä Plone-ympäristössä. Teemoitustyökalu tarjosi myös graafisen työkalun Diazo-sääntöjen luomiseen. Sen käyttö osoittautui hankalammaksi kuin itse sääntöjen käsin kirjoittaminen suoraan rules.xml-tiedostoon teeman muokkaustilan pääikkunassa. Sen sijaan teeman esikatselu toiminto koettiin hyödylliseksi, sillä sen avulla voitiin tarkastella teemaan luotuja muutoksia ilman jatkuvaa teeman pois ja päälle kytkemistä. Lisäksi yksittäisten tiedostojen tuonti sekä kansiorakenteiden luominen teemaan jälkikäteen onnistui tästä näkymästä. (Ks kuvio 13.)

Modify theme Oppariteema

[Back to the control panel](#)



Kuvio 13. Plonen teeman muokkausnäky

Plonen teemoitustyökalu mahdollisti opinnäytetyössä luodun teeman toteuttamisen ilman, että kehittäjän tarvitsi ymmärtää kaikkia Plonen sisältämiä ominaisuuksia. Toisaalta, jotta kaikki Plonen sisällönhallintatyökalut saatiin toimimaan muokatun ulkoasun kanssa, teemaan tuli sisällyttää myös Plonen vakiotyylitiedostot. Näistä sekä muista syistä aiheutuvia ongelmia käsitellään seuraavassa kappaleessa.

4.7 Ulkoasun implementointi Ploneen

Diazo-teema Plonessa on käytännössä ulkoasudemo, johon sisältö tuodaan rules.xml-tiedostossa määriteltyjen Diazo-sääntöjen sekä tarvittaessa XSLT-sääntöjen avulla. Teemaa luodessa sääntötiedostoon on oletuksena asetettu sääntöjä koskien seuraavia Plone-sivun sisältöjä:

- Teeman title korvataan Plonen omalla
- Base-elementin sisältö tuodaan Plonesta teemaan
- Lisätään Plonen head-tagien sisältö teeman head-elementin sisään
- Yhdistetään luokat Plonen ja teeman body-elementin sisällä
- Kopioidaan Plonen skriptit teeman body-elementin loppuun

Teemaa luodessa ensimmäiseksi ongelmaksi muodostuivat Plonen vakioulkoasun mukana tulevat tyylitiedostot. Perehtyminen CSS-selektorien tärkeysjärjestyksiin auttoi. Tärkeysjärjestys korkeimmasta prioriteetista matalimpaan on seuraava:

1. !important-määrite
2. style-määritteet itse HTML-elementissä
3. ID-määritteiden määrä
4. Muiden attribuuttien ja luokkien määrä

Mikäli elementillä on kaksi samanarvoista määritettä eri tyyli tiedostoissa, viimeisimpänä linkitetyn tyyli tiedoston määrite jää voimaan. Tämän takia päädyttiin muokkaamaan Plonen head-tagien sisällön tuontia siten, että Plonesta tuodut sisällöt ilmoitetaan ennen teemassa asetettuja sisältöjä. Näin teeman omat tyyli tiedostot tulevat viimeisenä ajaen yli Plonen samanarvoiset tyyli säännöt. (Calculating a selector's specificity 2011.)

Plonen omat tyyli tiedostot sai asetettua teeman tyyli tiedostoja ennen seuraavalla Diazo-komennolla:

```
<after css:theme="html head title"
content="/html/head/link | /html/head/style | /html/head/meta"
/>
```

After-komento asettaa tällöin teemassa olevan title-elementin perään Plonen omassa ulkoasussa olevien link-, style- ja meta-elementtien sisällöt. Koska teeman omat tyyli tiedostot ovat title-elementin alapuolella, ne tulevat vasta Plonen omien tyyli tiedostojen jälkeen.

EOTO-sivustolla on käytettävissä englannin ja suomen kieli. Koska kielivalikko sisältyy teemaan, sen kieli täytyy vaihtaa Diazoa hyödyntäen. Sivuston content-, eli sisältö-elementti saa luokan site-en Plonelta kielen vaihtuessa englanniksi. Luokan avulla voidaan luoda ehtolause kielimenun otsikon korvaamista varten. Tämä on toteutettu alla olevan ohjelmoinnin mukaisesti.

```
<replace css:theme=".check-lang" css:if-content="body.site-en">
  <a class="check-lang" href="#">LANGUAGE</a>
</replace>
```

Teemassa asetettiin myös kieli sisällölle html-elementin sisällä, sen korvaaminen onnistuu Diazolla. Koska sekä teemassa että Plonessa kyseiselle elementille on määritetty lang-, eli kieli-attribuutti, se voitiin kopioida Plone-sisällöstä teemaan seuraavalla tavalla:

```
<copy attributes="lang" css:theme="html" css:content="html" />
```

Sivustolle kirjautumiseen tarvittava työkalu tuotiin Plonesta teemaan alla olevan säännön avulla. Teemassa sijaitseva elementti, jonka luokka oli plonetools, korvattiin Plonessa sijaitsevalla elementillä, jonka id-attribuutin arvo oli portal-personaltools.

```
<replace css:content="#portal-personaltools"
css:theme=".plonetools" />
```

Navigaation generoiminen teemaan Plonen pohjalta loi uuden ongelman. Diazo on toiminnallisuudeltaan yksisuuntainen, eli sen avulla voidaan tuoda sisältöä Plonesta teemaan, mutta ei teemasta Ploneen. Teeman navigaatio korvattiin Plonen omalla, jotta voitiin säilyttää sivuston navigaation muokattavuus. Samalla se tuhosi navigaation li-elementtien luokan menu-item. Korvaus oli kuitenkin tehtävä. Se aloitettiin Diazo-säännön mukaisesti valitsemalla Plonen li-elementit, joiden id-attribuutin arvo on portal-globalnav. Ne sijoitettiin teemassa sijaitsevan elementin, jonka luokka oli menu, lapsielementtien sijaan.

```
<replace css:content="#portal-globalnav li"
css:theme-children=".menu" />
```

Jotta Plonesta tuotujen li-elementtien luokkiin voitiin lisätä teemassa olleet luokat, ne täytyi luoda XSLT:n, eli Extensible Stylesheet Language Transformations merkintäkielen avulla, johon Diazo pohjautuu. Rules.xml-tiedostoon on myös mahdollista kirjoittaa sääntöjä sen avulla. Ensimmäinen etsittiin Plonen sisällöstä kaikki ul-elementit, jonka id-attribuutin arvo oli portal-globalnav, lapsi li-elementit. Kaikki tulevat komennot suoritettiin siis jokaiselle lapsi li-elementille. Jokaisen elementin kohdalle asetettiin muuttuja nimeltä active. Mikäli li-elementille oli määritelty luokka nimeltä selected, tallennettiin muuttujan arvoksi " active". On olennaista, että sanaa active ennen tallennettiin välilyönti, sillä active-luokka lisättiin toisen luokan perään. Tämän jälkeen kopioitiin navigaation sisältö sellaisenaan lapsielementteineen, paitsi korvattiin jokaisen li-elementin luokan arvo arvolla menu-item. Menu-item-arvon perään liitettiin myös aiemmin tallennetun active-muuttujan sisältö, joka oli joko tyhjä tai sisälsi arvon " active".


```

<xsl:template match="//ul[@id='portal-globalnav']//li">
  <xsl:variable name="active">
    <xsl:if test="@class='selected'"> active</xsl:if>
  </xsl:variable>
  <xsl:copy>
    <xsl:copy-of select="@*" />
    <xsl:attribute name="class">
      <xsl:text>menu-item</xsl:text>
      <xsl:value-of select="$active" />
    </xsl:attribute>
    <xsl:copy-of select="node()" />
  </xsl:copy>
</xsl:template>

```

Plonen sisällönhallintapalkin tuominen teemaan oli yksinkertaisempaa, kun se sisällytettiin teemaan alla olevalla Diazo-säännöllä. Sivun sisällöstä otettiin id-attribuutin arvolla edit-bar merkitty elementti sisältöineen. Se liitettiin teemassa menu-luokan sisältävän elementin jälkeen.

```

<after css:theme=".menu" css:content="#edit-bar" />

```

Sisällönhallintapalkki oli tärkeä, sillä ilman sitä myöskään muihin sisällönhallintatyökaluihin ei olisi pääsyä. Seuraavaksi teemaan tuotiin Plonen pääsisältöelementti. Se oli merkitty Plonessa id-attribuutin arvolla content. Kyseisen elementin sisältö sijoitettiin teemassa main-luokan sisältämän elementin lasten paikalle.

```

<replace css:content="#content" css:theme-children=".main" />

```

Sääntöä luodessa huomattiin, että Plonen sisältöelementti vaikutti sivuston kirjautumistoimintoon. Mikäli pelkästään id-arvolla content merkityn elementin lapset olisi liitetty teemaan, kirjautumisvalikkoon sidottu sivuston päälle aukeava kirjautumisnäkyvä lakkasi toimimasta.

Alatunnisteen tuominen Plonesta teemaan toteutettiin seuraavalla Diazo-säännöllä. Teeman footer-elementin sisältö korvattiin Plonen portaalin vastaavalla.

```

<replace css:content="#portal-footer p" css:theme="footer p" />

```

Lopputuloksena saatiin teemaan Plonesta kaikki tarvittava sisältö. Alla oleva kuvio 14 esittelee teemoitetun sivuston etusivun. Sivustolle kirjaututtiin sisään sisällönhallintatyökalujen esittelemiseksi.

EOTO KIELI ADMIN ▼

Each One Teach One

Pääsivu Info Ilmoitukset Ideat & materiaalit Blogi

Sisältö **Näytä** Muokkaa Käyttöoikeudet Esitystapa ▼ Lisää uusi... ▼ Tila Julkaistu ▼

Etusivu

tekijä: admin — Viimeisin muutos maanantai 09. maaliskuuta 2015, 11.10 — [Historia](#)

Each One Teach One -ohjelma eli EOTO on kolmen opintopisteen laajuinen ohjelma sinulle, joka haluat tutustua erimaalaisiin opiskelijoihin ja oppia vieraita kieliä. Ohjelmassa opiskelijat opettavat toisilleen omaa äidinkieltään tai koulukieltään ja opiskelevat parinsa kieltä. Parityöskentelyn lisäksi kieliä voidaan opiskella myös pienissä ryhmissä, ja opiskelijat voivat perustaa myös yhteen kieleen keskittyviä keskusteluryhmiä ja lukupiirejä.

Jyväskylän yliopiston kielikeskus, PL 35 (Oppio), 40014 Jyväskylän yliopisto. (c) Kielikeskus, kaikkinaisen lainaaminen ilman kielikeskuksen lupaa kielletty.

 **Language Centre**
UNIVERSITY OF JYVÄSKYLÄ

 **JYVÄSKYLÄN YLIOPISTO**
UNIVERSITY OF JYVÄSKYLÄ

Kuvio 14. Sivuston teemoitettu etusivu

5 Tulokset ja pohdinta

Opinnäytetyön tarkoituksena oli luoda uusi ulkoasu Each One Teach One -sivustolle tutustuen samalla Plone-sisällönhallintajärjestelmään, responsiiviseen verkkosuunnitteluun sekä Diazo-teemoitukseen. Lopputuotteena saatiin Ploneen asennettava Diazo-teema. Opinnäytetyön kirjallisessa osassa käytiin läpi Plonen perustoiminnot, responsiivisen verkkosuunnittelun periaatteet sekä Diazo-teemoituksen käyttö.

EOTO-sivuston varsinainen ulkoasu-uudistus käynnistyy vasta syksyllä 2015, joten teeman viimeistelyä jatketaan myös opinnäytetyön jälkeen. Opinnäytetyössä keskityttiin sivuston ulkoasun uudistamiseen kurssin suorittajan näkökulmasta, joten myös hallinnoijan ja opettajien näkökulmaan pureutuminen tulee esiin jatkokehityksen yhteydessä.

Asiakas oli tyytyväinen ulkoasussa toteutettuihin muutoksiin ja yhteydenpito pysyi mukavan ripeänä sähköpostin sekä tapaamisien muodossa. Varsinaisen käyttöönoton

ollessa vielä edessä palaute on lähinnä rajoittunut Jyväskylän yliopiston IT-palveluiden Kehittämispalvelun sekä asiakkaan antamiin kommentteihin. Hieman ahkerampi yhteydenpito toimeksiantajan kanssa olisi nopeuttanut ulkoasun kehitystä, tämä oli tärkeä huomio kehittäjälle itselleen. Jatkossa kannattaa olla itse aktiivisempi.

Responsiiviseen verkkosuunnitteluun sekä teeman toteuttamiseen perehtyminen syventi ymmärrystä verkkokehittämisestä. Uusia tuttavuuksia olivat Plone-sisällönhallintajärjestelmä sekä Diazo-teemoitus. Vanhastaan tuttujen tekniikoiden ymmärrystä päästiin syventämään jQueryn, XSLT:n sekä HTML- ja CSS-formaattien muodossa.

Prosessia hidasti vanhan EOTO-sivuston vanhentunut Plone-versio, jonka takia uuden ulkoasun toteuttamiseen käytetyt rakenteet tuli luoda uudelleen ulkoasun esittelemistä varten. Toisaalta kyseinen ongelma pakotti tutustumaan syvemmin Plonen toiminnallisuuteen antaen paremman käsityksen sen toiminnoista.

Diazo-tekniikka osoittautui helposti omaksuttavaksi tavaksi luoda teema Plone-sisällönhallintajärjestelmään. Sääntöjen rakentuminen XML-kieleen teki sisäistämisestä helppoa. Diazon avulla voitiin tuoda sisältöä Plonesta teemaan, mutta toisin päin toimiminen ei ollut mahdollista. Toisaalta tämä rajoitus helpotti sääntöjen luomisprosessia, mutta esimerkiksi navigaatiota korvatta se aiheutti ongelmia. Ongelmat oli mahdollista ratkaista XSLT-merkintäkielellä, jonka luomiseen Diazo perustuu.

Ulkoasun tyylittely Plonessa osoittautui työlääksi, sillä Plonen elementit olivat raskaasti tyyliteltyjä ja niiden korvaaminen omasta tyylitiedostosta vei paljon aikaa. Jatkon kannalta saattaa olla tarpeen käydä läpi Plonen hallintapaneelista päälle laitettut tyylit.

jQueryn hyödyntäminen tämän opinnäytetyön yhteydessä jäi pintaraapaisun tasolle. Jatkokehitys saattaa tuoda sille lisää käyttökohteita, tai vaihtoehtoisesti paljastaa sen tarpeettomaksi. JQueryn tarpeellisuus riippuu jatkossa siitä, kuinka paljon sisällön rakennetta luodaan Plonen omilla työkaluilla sivuston käyttöönottoa valmistellessa.

Lähteet

About Plone 2015. Plonen verkkosivut. Viitattu 4.5.2015. <http://plone.com/about>

Adding New Content 2015. Plonen dokumentaatio. Viitattu 4.5.2015. <http://docs.plone.org/working-with-content/adding-content/adding-new-content.html>

Bicking, I. 2012. The Philosophy Of Deliverance. Viitattu 4.5.2015. <https://pythonhosted.org/Deliverance/philosophy>

Bracey, K. 2015. The State of Responsive Web Design. Tuts+ verkkosivu, Web Design. Viitattu 4.5.2015. <http://webdesign.tutsplus.com/articles/the-state-of-responsive-web-design--cms-22878>

Calculating a selector's specificity. 2011. World Wide Web Consortium. Viitattu 4.5.2015. <http://www.w3.org/TR/2011/REC-CSS2-20110607/cascade.html#specificity>

Callagher, N. 2012. About normalize.css verkkosivu. Viitattu 4.5.2015. <http://nicolasgallagher.com/about-normalize-css/>

Conceptual Overview 2015. Plonen dokumentaatio. Viitattu 4.5.2015. <http://docs.plone.org/working-with-content/introduction/conceptual-overview.html>

Introduction to Collections 2015. Plonen dokumentaatio. Viitattu 4.5.2015. <http://docs.plone.org/working-with-content/using-collections/newstyle/introduction-to-collections.html>

jQuery 2015. Verkkosivun etusivu. Viitattu 4.5.2015. <https://jquery.com>

Luckow, S. & Frank, W. 2013. Cms Gardener's Guide, Deciding On Open Source. Berliini: Drupal-Community e.V.

Marcotte, E. 2009. Fluid Grids. A List Apart verkkosivu. Viitattu 4.5.2015. <http://alistapart.com/article/fluidgrids>

Marcotte, E. 2010. Responsive Web Design. A List Apart verkkosivu. Viitattu 4.5.2015. <http://alistapart.com/article/responsive-web-design>

Marcotte, E. 2011. Fluid Images. A List Apart verkkosivu. Viitattu 4.5.2015. <http://alistapart.com/article/fluid-images>

plone.app.theming manual. 2015. Plonen dokumentaatio. Viitattu 4.5.2015. <http://docs.plone.org/external/plone.app.theming/docs/index.html>

Plone User Accounts and Roles 2015. Plonen dokumentaatio. Viitattu 4.5.2015. <http://docs.plone.org/working-with-content/introduction/user-accounts-and-roles.html>

Quickstart 2015. Plonen dokumentaatio. Viitattu 4.5.2015. <http://docs.plone.org/quickstart>

What is Plone? 2015. Plonen dokumentaatio. Viitattu 4.5.2015.
<http://docs.plone.org/intro>