

OPINNÄYTETYÖ

Satu Micklin 2010

**RITAVALKEA.COM – SIVUSTON
UUDISTUS PÄIVITETTÄVYYDEN
NÄKÖKULMASTA**



Rovaniemen
ammattikorkeakoulu
University of Applied Sciences

TIETOJENKÄSITTELYN KOULUTUSOHJELMA

ROVANIEMEN AMMATTIKORKEAKOULU

KAUPPA JA HALLINTO

Tietojenkäsittelyn koulutusohjelma

Opinnäytetyö

**RITAVALKEA.COM – SIVUSTON UUDISTUS
PÄIVITETTÄVYYDEN NÄKÖKULMASTA**

Micklin Satu

2010

Ritavalkean laskettelukeskus ja Pellon kunta

Ohjaaja Jortikka Aarre

Hyväksytty _____ 2010 _____

Tekijä	Satu Micklin	Vuosi	2010
Toimeksiantaja	Ritavalkean laskettelukeskus ja Pellon kunta		
Työn nimi	Ritavalkea. com – sivuston uudistus		
Sivu- ja liitemäärä	28 + 4		

Opinnäytetyöni tehtävänä oli uudistaa Ritavalkean laskettelukeskuksen verkkosivut. Ritavalkean entisten kotisivujen koettiin vanhentuneen, eivätkä ne erottuneet positiivisesti muista sivustoista.

Tavoitteena oli luoda sivut, jotka olisivat selkeät ja helppo päivittää.

Sivuston uudistuksessa käytettiin XHTML- ja CSS- kieliä.

Opinnäytetyöprosessin tuloksena saatiin täysin uudistetut www-sivustot, jotka on helppo päivittää.

Tutkimusongelmanani oli helppo päivitettävyyden, minkä takaamiseksi asiakkaalla on mahdollista saada koulutusta niin halutessaan. Koulutusta ei kuitenkaan käsitellä tässä opinnäytetyössä.

Author	Micklín Satu	Year	2010
Commissioned by	Ritavalkea Ski Centre and the municipality of Pello		
Subject of thesis	Reformation of web pages		
Number of pages	28 + 4		

The main purpose of this thesis was to reform the web pages for Ritavalkea Ski Centre. Their current pages were out of date and they did not stand out positively when compared to other pages.

The main goal was to create pages that are uncluttered and easy to update.

The web pages were reformed using XHTML and CSS.

As a result of this thesis work: the web pages for Ritavalkea Ski Centre have been completely reformed.

Also, since the main point of the thesis was to create web pages with easy updating, this could be demonstrated if the Ritavalkea Ski Centre as the client wanted to have a tutorial. However, the tutorial is not included in this thesis.

Key words

CSS, XHTML, website

SISÄLTÖ

1 JOHDANTO	1
2 KÄYTETYT TEKNIIKAT JA OHJELMAT	3
2.1 HTML & XHTML.....	3
2.2 CSS.....	5
2.3 KÄYTETYT OHJELMAT	7
3 LÄHTÖTILANNE	10
3.1 TEKNISET ONGELMAT	10
3.2 VANHA ULKOASU.....	11
3.3 KÄYTETTÄVYYS.....	13
4 UUDEN SIVUSTON MAHDOLLISUUDET JA RAJOITUKSET	14
5 UUSI SIVUSTO	15
5.1 SIVUSTON UUDISTUKSET	15
5.2 TEKNINEN TOTEUTUS.....	18
5.4 GOOGLE ANALYTICS.....	21
5.5 TESTAUS.....	22
6 PÄIVITETTÄVYYS	24
7 YHTEENVETO	27
LÄHTEET	28
LIITTEET	29
LIITE 1: TYYLITIEDOSTO.....	29

1 JOHDANTO

Aiheen opinnäytetyölleni löysin, kun toimeksiantaja otti yhteyttä kouluuni. He tarvitsivat kotisivuilleen perusteellisen uudistuksen ja ennen kaikkea sivut, jotka olisi helppo päivittää. Toimeksiantajana työssäni toimii Ritavalkean laskettelukeskus ja Pellon kunta. Laskettelukeskuksen historia on pitkä ja nykyiselle se on rakennettu vuosien saatossa.

Pellon kunta rakensi vuosina 1977 ja 1978 leirikeskukseen Ritavalkealle nuorisotoimen käyttöön ja Pellon Laskettelurinteet Oy perustettiin 1981. Perustetun yhtiön hallitus päätyi kustannussyistä rakentamaan laskettelurinteiden Lupovaaraan. Siltä varalta, että toimintaa laajennettaisiin myöhemmin Ritavalkealle, otettiin yhtiön nimeksi Pellon Laskettelurinteet Oy. Lupovaaran rakentaminen toteutettiin vuosina 1982 ja 1983. Nykyinen Ritavalkea rakennettiin vuosien 1988 ja 1990 välisenä aikana.

Pellon Laskettelurinteet Oy:n liikevaihto ja asiakasmäärät olivat suurimmillaan Ritavalkean viiden ensimmäisen toimintavuoden aikana. Matkailijoiden määrä on viime vuosina ollut suurempi kuin toiminnan alkuvaiheessa ja se on ollut jatkuvasti pienessä kasvussa.

Nykyisin jokaisella yrityksellä on kotisivut ja ihmiset olettavat löytävänsä kaiken informaation yritysten kotisivuilta. Kotisivuja tulisi myös päivittää tarpeeksi usein, jotta ihmisten mielenkiinto säilyy eivätkä he siirry toisten yritysten pariin.

Ritavalkealla on ollut kotisivut jo useamman vuoden, mutta nykytilassaan ne eivät kohtaa asiakkaiden odotuksia. Yritys toivookin toimivien kotisivujen kautta saavansa laajemman asiakaskunnan. Yrityksessä ei kuitenkaan ole ketään, jolla olisi laajempi kokemus tietokoneista tai mistään ohjelmointikielistä. Tästä löysin opinnäytetyöhöni tutkimusongelman: miten tehdä koti-

sivut, joita voi päivittää henkilö, jolla ei ole mitään kokemusta ohjelmoinnista eikä edes englannin kielen osaamista?

Kaikkein luonnollisin vaihtoehto olisi ollut toteuttaa sivut jonkin julkaisujärjestelmän (Joomla, Wordpress) avulla. Tämä ei kuitenkaan ollut mahdollista palvelimen tuomien rajoitusten myötä, joten päädyin toteuttamaan sivut XHTML-kieltä ja CSS – tyyliohjeita käyttäen. Itse en ole käyttänyt XHTML-kieltä koskaan ja CSS-kielen olen tutustunut vain lyhyesti työharjoitteluni aikana, joten tätä kautta koin oppivani uutta ja saavani lisähaastetta opinnäytetyöhöni.

2 KÄYTETYT TEKNIIKAT JA OHJELMAT

2.1 HTML & XHTML

Julkaisukielet ovat merkintäkieliä, jotka kuvaavat dokumentin rakennetta. Julkaisukielet sisältävät elementtejä eli komentoja, jotka määrittelevät esimerkiksi otsikon paikan sivulla. Elementit määritellään kulmasuluin ja useimmat niistä toimivat pareittain, niin että aloittava merkintä asettaa komennon voimaa ja päättävä merkintä lopettaa sen. (Keränen-Lamberg-Penttinen 2003, 8.)

HTML eli HyperText Markup Language on yksinkertainen ja helposti opittava julkaisukieli, joka koostuu sivun rakennetta kuvaavista elementeistä sekä niiden attribuuteista. HTML-kielen kehityksestä on vastannut W3C, mutta myös selainvalmistajat ovat tehneet omia ratkaisujaan selainten tulkitsemaan HTML-kielen. (Keränen ym. 2003, 8.)

HTML-dokumentti on tekstimuotoinen dokumentti, joka sisältää tekstiä ja HTML-merkkejä. HTML-dokumentin tiedostopääte on yleensä .html tai .htm. (Korpela-Linjama 2004, 11.)

HTML-dokumentissa pakollisia osia on dokumenttityyppi ja TITLE-elementti, mutta <html>, <head> ja <body>-elementtien käyttäminen on suositeltavaa. Kuvassa 1 on esimerkki HTML-dokumentin perusrakenteesta.

```
<html>
  <head>
    <title>Esimerkki</title>
  </head>
  <body>
    <p>Tähän tulee tekstiä</p>
  </body>
</html>
```

Kuvio 1. Esimerkki HTML-dokumentin perusrakenteesta

XHTML on HTML- kielen kaltainen julkaisukieli, jossa on kaikki HTML- kielen elementit ja ominaisuudet, mutta elementtejä käytetään XML- kielen säännöillä. HTML- kielestä poiketen XHTML- kielessä on tarkkaan säännelty kirjoitustapa. (Keränen ym. 2003, 9.)

XHTML vaatii suurempaa järjestelmällisyyttä merkkauksen eräissä yksityiskohdissa. Esimerkiksi jokaisella elementillä on päättävä merkintä ja elementtien nimet kirjoitetaan pienin kirjaimin. XHTML- kielessä myös attribuuttien arvot on kirjoitettava lainausmerkeillä, vaikka HTML- kielessä ne voi jättää pois. (Korpela-Linjama 2004, 28.)

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>TÄMÄ ON OTSIKKO</title>
</head>
<hr />
<body>
</body>
</html>
```

Kuvio 2. Esimerkki XHTML- dokumentin perusrakenteesta

Yleisellä kappaletason elementillä eli div- elementillä voidaan merkitä erottuvia kokonaisuuksia asiakirjasta, esimerkiksi ylätunniste tai navigointi. Sitä voidaan käyttää kappaleetasolla kun muuta sopivaa elementtiä ei löydy. (Haverinen 2010.)

```
<div id="kokosivu">
<div id="sisalto">
<p>tekstiä</p>
</div>
</div>
```

Kuvio 3. Esimerkki div- elementin käytöstä

2.2 CSS

CSS- kielellä (Cascading Style Sheets) laaditaan dokumentin ulkoasua kuvaavia tyyliohjeita. Tyyliohjeiden avulla sivuston ulkoasua ei tarvitse lisätä www-sivun rakenteeseen ja muutoksia tehdessä muutokset kohdistuvat kaikkiin sivuihin, joissa tyyliohjeita käytetään. Kaikki sivun ulkoasun ja typografian määrittelyt, esimerkiksi tekstin koko ja väri, tehdään tyyliohjeeseen. Näistä säännöistä voi koota luokan, johon viittaamalla saadaan tyyli käyttöön. (Keränen ym. 2003, 11.)

Seuraavassa esimerkissä on tyyli luokka, jossa määritellään tekstin kirjasin, koko ja tyyli.

```
.teksti
{
font-family:Arial, Helvetica, sans-serif;
font-size:12px;
font-style:italic;
}
```

Kuvio 4. Esimerkki tyyli luokasta

Tyyliohje voi myös olla osana HTML- dokumenttia, mutta sen muoto ja merkitys eivät määräydy HTML- kielen mukaan. CSS- kielen ilmaisujen perusrakenne on kuitenkin yksinkertainen. Halutessaan voi vain antaa ominaisuudelle arvon kirjoittamalla ominaisuuden nimi, kaksoispiste ja arvo. (Korpela-Linjama 2004, 8.)

```
text-align: right;
```

Kuvio 5. Esimerkki yksinkertaisesta CSS- koodista

Esimerkki ei kuitenkaan kerro, minkä kohteen ominaisuus tässä asetetaan. Tilanne olisi kuitenkin täysin eri tehtäessä pieni muutos:

```
td
{
text-align: right;
}
```

Kuvio 6. Edellisen kuvan koodi muokattuna

Esimerkkiin lisättiin td, mikä käytännössä tarkoittaa, että esimerkissä asetetaan yhdellä kertaa kaikkien td- elementtien text-align – ominaisuus. (Korpela-Linjama 2004, 9.)

Kuvan 3 div-elementin ominaisuuksia voisi CSS- kielellä muokata esimerkiksi seuraavalla tavalla:

```
#sisalto
{
width:100px;
float:left;
}
```

Kuvio 7. Div- elementin muokkausta CSS- kielellä

CSS- kielessäkin on kuitenkin ongelmansa, etenkin ulkoisen tyylitiedoston käytön kanssa. Vaikka haluaisit tehdä vain pienen muutoksen, täytyy uusi tyylitiedosto luoda, linkittää se dokumenttiin ja mahdollisesti luomaan pseudoluokkiakin. Suurin hankaluus CSS- kielen käytössä on ollut sen vaihteleva tuki eri selaimilla. Etenkin Internet Explorer- selainta varten on aiemmin jouduttu lisäämään erillisiä määrittelyjä. (Taavetti 2007.) Uudistamassani sivustossa ei kuitenkaan selainten kanssa ole ilmennyt ongelmia, sillä nykyisin selainten tuki CSS- kielelle on parantunut, ja olen pyrkinyt pitämään koodin mahdollisimman yksinkertaisena. Toimeksiantaja ei myöskään ole tekemäs-

sä muutoksia CSS- tiedostoon, joten ulkoisen tyyli-tiedoston käyttäminen ei tässä tapauksessa ole ongelma.

Halusin käyttää sivuston uudistuksessa juuri näitä kieliä, sillä CSS- ja XHTML- kielien olemassaolon perimmäisiä syitä on halu eriyttää sivuston sisältöön liittyvä merkintä ulkoasuun liittyvästä merkinnästä. Ensimmäinen hyöty saavutetaan siistimpänä HTML / XHTML- merkintänä, joka kuvaa vain dokumentin rakennetta. Tällainen merkintätapa tarkoittaa usein pienempää koodimäärää, joka näkyy helpompana koodin hallintana, jolloin päivittäminen helpottuu, sekä nopeampina latausaikoina. Toinen hyöty saavutetaan, kun sivuston ulkoasua halutaan päivittää, sillä koko sivuston ulkoasu on mahdollista päivittää ainoastaan CSS- tiedostoa muuttamalla. (Koulutus- ja konsultointipalvelu KK Mediat 2010.)

2.3 Käytetyt ohjelmat

Heti alusta alkaen yritin miettiä parasta vaihtoehtoa koodin luontiin. Aluksi ajattelin käyttäväni pelkkää Notepadia, mutta koska en tuntenut XHTML- kieltä ennestään, tuntui se turhaan työläältä. Kokeilin siis myös muita vaihtoehtoja. Ensimmäinen vaihtoehto olisi ollut Crimson Editor V3.72, mutta koin sen lähes yhtä työlääksi kuin Notepadinkin. Seuraava vaihtoehto olisi ollut KompoZerin WYSIWYG- editori. ”WYSIWYG” on lyhenne lauseesta ”What you see is what you get”, eli ”se mitä näet, sen saat”. Kyseistä lyhennettä käytetään ohjelmistoista, joilla muokattaessa www-sivun sisältö näyttää hyvin samalta kuin lopputulos (ATPM 1998). Valitettavasti KompoZerilla ja käyttöjärjestelmälläni oli yhteensopivuusongelmia, joten jouduin hylkäämään tämänkin vaihtoehdon. Tämän jälkeen kokeilin vuorostani Macromedia Dreamweaveria. Testikäytössä ohjelma oli todella helppokäyttöinen, ja olisin mielelläni tehnyt sillä sivut. Kuitenkin kooditasolla ohjelma tuotti paljon ylimääräistä tavaraa ja olisi vaatinut Dreamweaver- ohjelman käyttöä helpon päivitettävyyden saavuttamiseksi. Kuitenkin toiveena nimenomaan oli, ettei toimeksiantaja joutuisi käyttämään mitään maksullista ohjelmaa sivustoa päivitettäessä, joten Dreamweaver ei sopinut tähän käyttötarkoitukseen.

Viimeisenä vaihtoehtona oli Microsoft Office SharePoint Designer 2007. Olin heti alussa hylännyt sen, sillä olen käyttänyt sitä aiemmin töissä ja kokenut sen hyvin työlääksi ohjelmaksi. Olen myös kuullut siitä paljon kritiikkiä, mikä vielä entisestään nosti kynnystä kokeilla kyseistä ohjelmaa. Kuitenkin kokeiltuani ohjelmaa huomasin, että se soveltuu parhaiten tällaisen työn toteuttamiseen. Ohjelmassa on mahdollista käyttää rakennenäköä, mikä käytännössä vastaa WYSIWYG- editoria, mutta koin sen tarpeettomaksi ja käytin vain koodi- näköä. Sen avulla pystyin itse täysin säätämään koodia ja tekemään siitä mahdollisimman yksinkertaisen. Koodia kirjoittaessa SharePoint Designer ilmoittaa puuttuvista sulkeista ja jos esim. elementiltä puuttuva lopettava merkintä. Tämä ominaisuus esti kaikki kirjoitus- ja huolimattomuusvirheet, mikä oli iso etu. Suurin etu oli kuitenkin se, että koodia kirjoitettaessa ohjelma automaattisesti tarjoaa eri vaihtoehtoja, mikä jälleen vähentää kirjoitusvirheiden syntymistä ja tarjoaa kokemattomalle XHTML- kielen käyttäjälle suuren avun. Kuvassa 8 näkyy kyseinen ”apuvalikko” auenneena. Hakuvaihtoehtoja voi tarkentaa kirjoittamalla esimerkiksi halutun elementin ensimmäinen kirjain, jolloin sen voi selata valikosta ja lisätä klikkaamalla. Ohjelma lisää tällöin sekä aloittavan että lopettavan merkinnän. Kuvassa näkyy auki XHTML- kielen valikko, mutta samanlainen tuki löytyy myös CSS- kielestä.

```

1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2 <html xmlns="http://www.w3.org/1999/xhtml">
3
4 <head>
5 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
6 <title>Nimetön 1</title>
7 </head>
8
9 <body>
10
11 <
12 h4
13 h5
14 h6
15 hr
16 i
17 iframe
18 img
19 input
20 ins
21 isindex

```

Kuvio 8. SharePoint Designerin apuvalikko

Sivustoa varten tein ulkoasun käyttäen ohjelmana Photoshop CS 2- ohjelmaa. Ohjelman valinta oli helppo, sillä Photoshop on tullut tutuksi sekä kou-

lun että yksityiselämänkin puolesta. Monimutkaisimmat ominaisuudet kyseisestä ohjelmasta eivät ole enää niin tuoreessa muistissa, mutta tarvitsinkin pääasiassa vain piirtotyökaluja, varjostusta sekä layereita, eli tasoja, jotta sain yhdisteltyä eri kuvia ja kuvien osia.

Tein aluksi Photoshopilla erilaisia kuvia mahdollisesta ulkoasusta ja sopivan vaihtoehdon löydyttyä muokkasin logon ja tein ”palaset” CSS- tiedostoa varten, eli tein taustaväriin ja jaoin sopivan taustan tarvittaviin osiin. Myös sivuille tulevia kuvia muokkasin Photoshopilla, pääasiassa säädin väritasoja, jotta kuvat olisivat saman sävyisiä ja kirkkaus vastaisi toisiaan. Lisäsin kuville sopivan taustan ja yhdistelin niitä tasojen avulla. Lisäsin vielä kuville varjon ja tallensin ne JPG- tiedostoiksi, jotta kuvat voisivat mahdollisimman vähän tilaa, mutta näkyisivät myös sivuilla halutulla tavalla.

3 LÄHTÖTILANNE

3.1 Tekniset ongelmat

Vanhassa sivustossa tekniset ongelmat olivat pienin murhe, mutta virheitä löytyi silti luvattoman paljon.

Vanhat sivut on tehty vuonna 2008 HTML- kielellä Microsoft FrontPagea apuna käyttäen. Sivut ovat siis kohtalaisen uudet, mutta alkujaankin ne on tehty vanhentuneiden standardien mukaan. Valitettavasti alkuperäistä versiota sivuista ei ole saatavilla, joten minulla ei mahdollisuuksia arvioida sivuston alkuperäistä tilaa. Sivustoon on kuitenkin tehty ajan kuluessa paljon muutoksia, sillä HTML- koodi on puuroutunut lähes tunnistamattomaan muotoon. Tämän vuoksi koodi onkin hyvin vaikeasti tulkittavaa.

Koodista huomaa helposti, että muutoksia on tehnyt henkilö, joka ei tunne HTML- kielen sääntöjä tai suositeltavaa kirjoitustapaa. Elementit on siroteltu ympäriinsä ja ulkoasu on määritelty HTML- kielellä koodiin ilman loogista järjestystä. Kommentointia ei ole käytetty, mikä osaltaan vaikeuttaa koodin tulkintaa ja sivujen päivittäminen olisi tässä vaiheessa hyvin vaikeaa.

Sivuille on myös lisätty kuvia virheellisillä nimillä, mikä tietenkin estää oikean kuvan näkymisen. Tiedostorakenne on tehty hankalaksi, sillä osa kuvista sijaitsee palvelimella /images- kansiossa ja osa taas on jätetty palvelimen juureen. Näkisin myös virheeksi sen, että /images- kansion sisältö palvelimella on kaikille avoin ja näin ollen kuka tahansa pääsee selaamaan kyseisen kansion sisältöä.

Vaikeuksia tuottaa myös se, että kuvia ja sivuja ei ole nimetty johdonmukaisesti. Samat kuvat löytyvät useilla eri nimillä ja sivuilta poistettua materiaalia löytyy eri paikoista, eikä esimerkiksi sille varatusta kansioista. Myös sivut au-

keavat eri paikkoihin, ensimmäinen saattaa aueta samaan ikkunaan ja toinen taas uuteen välilehteen. En osaa arvioida, että onko tämä tarkoituksella tehty ominaisuus, vai onko tullut virhe, josta tämä toiminto johtuu.

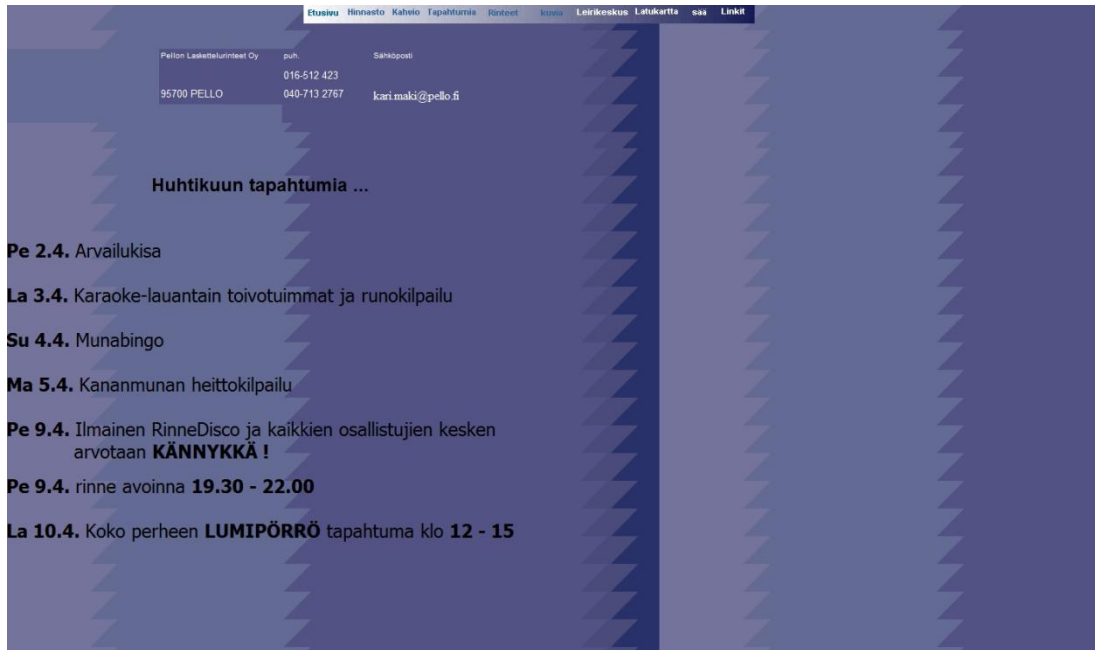
Sivustossa on myös käytetty kehyksiä, jotka toisinaan saattavat olla ihan toimiva ratkaisu. Kuitenkaan toisen sivun tuominen omiin kehyksiin ei ole hyvän tavan mukaista, puhumattakaan ongelmista tekijänoikeuksien kanssa.



Kuvio 9. Toisen sivuston tuominen omiin kehyksiin ei ole suositeltavaa

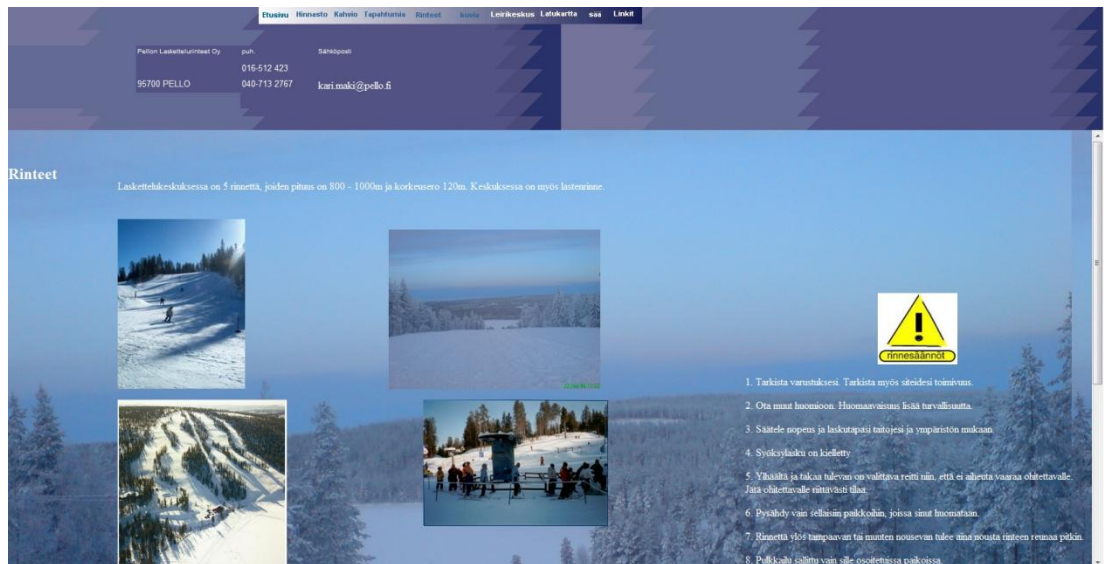
3.2 Vanha ulkoasu

Vanhaa ulkoasua voisin kutsua hankalaksi. Sivuston tausta oli violetti ja teksti toisinaan valkoisella, mustalla tai sinisellä. Navigaation tausta oli violetti ja siinä tekstit näkyivät valkoisella tai sinisellä.



Kuvio 10. Esimerkki Ritavalkean vanhasta ulkoasusta

Sivut eivät noudattaneet mitään graafista linjaa, sillä taustakuvat ja fontti saattavat vaihdella sivusta riippuen.



Kuvio 11. Esimerkki sivuston ulkoasun vaihtelusta

Sivuston ulkoasu on vanhentunut eikä pysty kilpailemaan nykyaikaisten sivujen kanssa. Myös markkinointi on unohtunut sivuja tehdessä.

3.3 Käytettävyys

Käytettävyys on jäänyt hieman vähemmälle huomiolle näitä sivuja tehtäessä. Navigaatio on suhteellisen pieni, mikä saattaa tuottaa ongelmia. Häiritsevää on myös jo aiemmin mainittu ikkunoiden aukeneminen eri ikkunoihin tai välilehtiin.

Tekstien ja kuvien sijoittelu vaihtelee sivusta riippuen, mikä vaikeuttaa sivujen hahmottamista. Hahmottamista vaikeuttaa myös ns. katseenkiinnittäjän puuttuminen. Myös teksti tuottaa ongelmia, sillä toisinaan sitä on vaikea erottaa taustasta, eikä tekstiä ole jäsennelty tai oikoluettu.

4 UUDEN SIVUSTON MAHDOLLISUUDET JA RAJOITUKSET

Sivustoa suunniteltaessa piti ensimmäisenä käydä läpi rajoitukset. Ensimmäinen vaihtoehtohan oli tehdä sivut julkaisujärjestelmällä helpon päivitettävyyden vuoksi, mutta toteutus ei ollutkaan mahdollinen. Vanhat sivut nimittäin sijaitsevat Pellon kunnan palvelimella, joten minulla ei ole pääsyä sinne. Palvelimen vaihto on todennäköisesti edessä tulevaisuudessa, mutta se ei valitettavasti tämän hetkistä tilannetta auttanut.

Päädyin lopulta XHTML- ja CSS- kielten yhdistelmään, sillä se asettaa tulevalle palvelimelle vähiten rajoituksia. Etuna on myös se, että CSS- kielen avulla saan tyyliohjeet eri tiedostoon, eivätkä ne vaikeuta sivujen päivitystä. Jouduin toki pelkistämään koodia ja lisäämään joukkoon kommentteja koodin selventämiseksi, sillä tuleva päivittäjä ei hallitse ohjelmointikieliä tai englantia.

Toimeksiantajalla ei ollut mielessä mitään tiettyä kohderyhmää, mutta itse ajattelin, että sivujen kohderyhmä koostuu pääasiassa peruskäyttäjistä, jotka etsivät tietoa Lapin laskettelukeskuksista ja joiden laitteistot koostuvat PC-järjestelmistä.

Ulkoasuun toimeksiantaja antoi rajauksen vain värien suhteen. Rajoittavana tekijänä tässäkin oli tulevan päivittäjän taidot. Sivuihin olisi voinut tehdä visuaalisesti paljon monipuolisemmat, mutta koodi täytyi pitää yksinkertaisena ja selkeänä. Lisäksi ei saanut asettaa rajoituksia tulevan palvelimen valinnalle ja kustannuksia ei saanut syntyä ollenkaan. Ja sivuja pitää pystyä päivittämään pelkän Notepadin avulla.

5 UUSI SIVUSTO

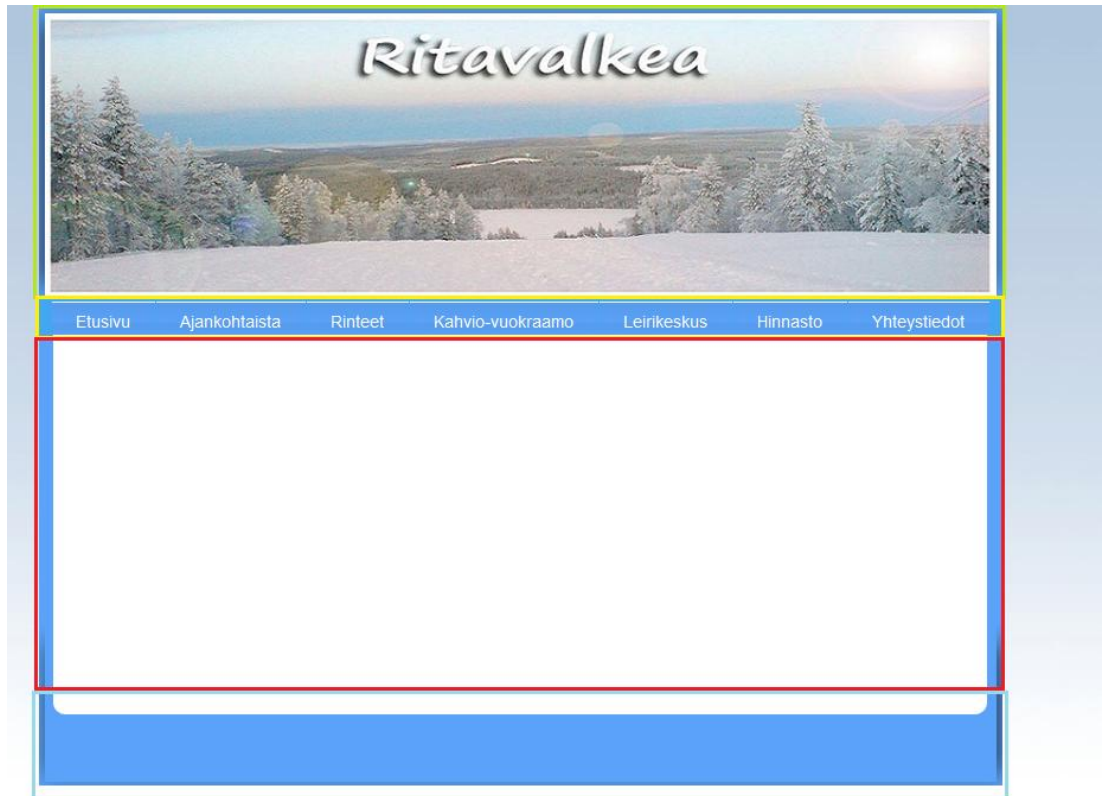
5.1 Sivuston uudistukset

Sivusto uudistettiin siihen pisteeseen, ettei vanhasta sivustosta jäänyt kuin joitain kuvia ja tekstejä käyttöön. Käyttökelpoista koodia ei vanhassa sivustossa ollut, joten aloitin kaiken puhtaalta pöydältä. Sivuston rakenne uusittiin kokonaan ja kehykset jätettiin pois käytöstä. Samalla sivujen sisältöä tiivistettiin, eli poistettiin toistot ja pyrittiin saamaan kaikki oleellinen tieto mahdollisimman tiiviiseen pakettiin.

Ulkoasu päivitettiin nykyaikaisemmaksi ja värimaailma muutettiin täysin. CSS-tyyliohjeet tehtiin erilliseen tiedostoon ja tuodaan jokaiseen XHTML-dokumenttiin, jotta jokaisen sivun ulkoasu olisi yhdenmukainen. Tällä saavutetaan paitsi tilan säästö dokumentissa, myös helppo päivitettävyyden (Jaakkola 2010).

XHTML-dokumentti sisältää mahdollisimman vähän elementtejä päivitettävyyden helpottamiseksi ja sivuston ainoa muuttuva osa on sisältö. Elementtejä on yläosa (header), navigaatio (nav), sisältö (main) ja alaosa (footer), jotka tehtiin div-elementteinä, joilla ei ole erityistä ulkoasua, mutta niitä voidaan käyttää kappaleetasolla kun muuta sopivaa elementtiä ei löydy (Haverinen 2010). Lisäksi on vielä wrapper, joka määrittelee palstojen määrän. Tässä tapauksessa palstoja tuli vain yksi, sillä vähäisen sivu- ja materiaalmäärän vuoksi ei ollut tarvetta esimerkiksi toiselle navigaatiolle, jonka olisi voinut sijoittaa toiseen palstaan.

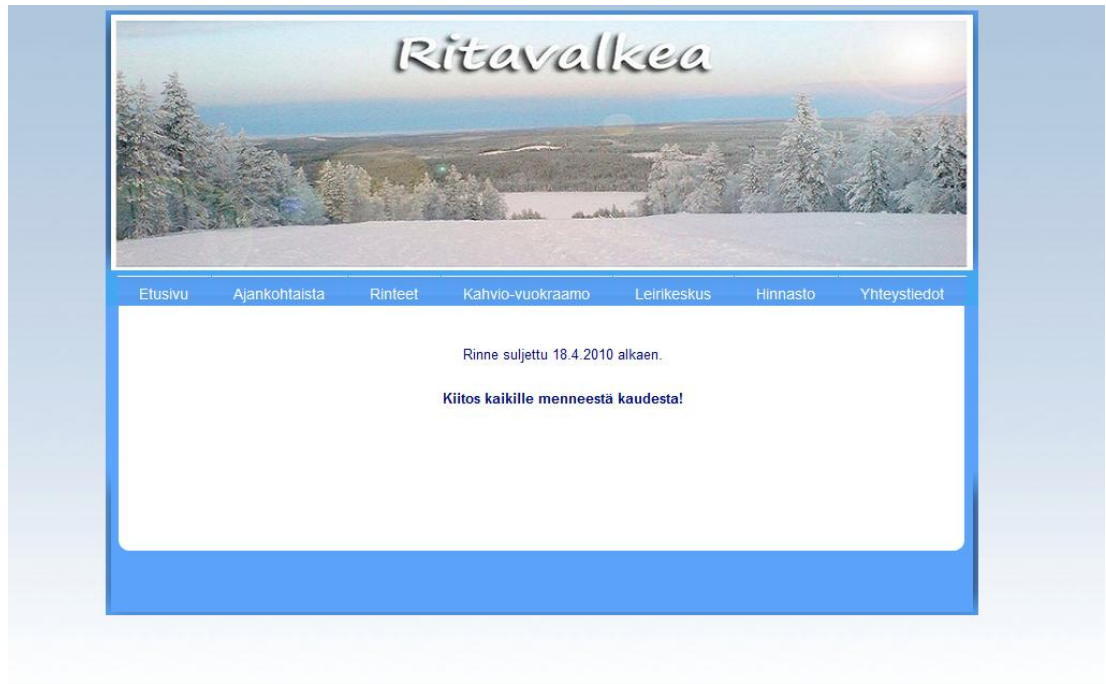
Seuraavalla sivulla oleva kuva 12 havainnollistaa sivuston ulkoasua. Kuvassa on sivuston elementit eri väreillä ympyröityinä.



Kuvio 12. Sivuston elementit ympyröityinä

Elementtien leveydet määriteltiin prosentteina, jolloin leveys lasketaan vapaana olevan tilan mukaan. Tällöin elementin leveys, joksi on määritetty 50 %, on tasan puolet ikkunan leveydestä ja muuttuu ikkunan koon vaihteluiden mukaan (Valkeinen 2010). Fonttikoko määriteltiin myös prosentteina, sillä pikseleitä käytettäessä fonttikoko olisi absoluuttinen, eikä skaalautuisi käyttäjän asetusten mukana (Penttinen 2008).

Ritavalkean laskettelukeskuksella ei ole käytössä mitään graafista ohjeistusta. Ainoa ohjeistus toimeksiantajalta oli, että sivuissa voisi olla sinistä ja oranssia, ja olisi hyvä, jos ne olisivat selkeät ja yksinkertaiset. Ilmeisesti Ritavalkealla ei ennestään ollut myöskään mitään painettua materiaalia, jotta olisi saanut sivuista graafisesti yhdenmukaiset muun materiaalin kanssa. Sain siis vapaat kädet ja onnistuin tekemään sivut, jotka ovat yksinkertaiset ja selkeät, mutta myös helppo päivittää.



Kuvio 13. Uuden sivuston ulkoasu

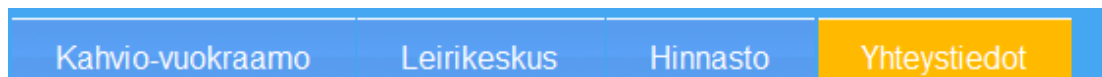
Ritavalkealla ei ollut ennestään logoa, jota olisin voinut hyödyntää sivustossa. Päädyin lopuksi muokkaamaan yhtä kuvista, jotta saataisiin jonkinlainen ”logo” sivustolle. Logo antaa yritykselle uniikin identiteetin ja luo mielikuvaa yrityksestä (Bonweb 2010).



Kuvio 14. Ritavalkean uusi ”logo”

Navigaation avulla käyttäjä pääsee liikkumaan eri sivujen välillä. Navigaation linkit tehtiin muistuttamaan painikkeita, jotka erotetaan toisistaan ja taustasta reunaviivoilla. Perustilassa painikkeet ovat sinisävyiset ja hiiren osuessa kohdalle aktiivinen linkki muuttuu oranssiksi. Navigointi on toteutettu

XHTML:llä ja muotoiltu CSS:llä. Navigaation teossa olisi voinut käyttää taulukkoa, mutta tällä tavalla päivittäminen ja muutoksien tekeminen helpottuu huomattavasti. Vältin taulukoiden käyttöä muutenkin sivustossa, sillä taulukon rakenne on XHTML- elementeistä monimutkaisin (Haverinen 2010). Käytin navigaation tekemiseen valmista rakennetta, sillä nettisivujen ulkoasun suunnittelussa ei ole järkevää lähteä keksimään pyörää uudestaan (MVnet 2010). Sopiva navigaatio löytyi Servage Magazinen blogista osoitteesta: <https://www.servage.net/blog/2009/03/20/create-a-cool-css-based-drop-down-menu/>.



Kuvio 15. Osa navigaatiosta taustaa vasten

5.2 Tekninen toteutus

Päivittäjän näkökulman lisäksi sivustoa tehdessä täytyy myös miettiä käyttäjän näkökulmaa. Apuna käyttäjän näkökulman miettimisessä käytin Penttisen (2010) tarkistuslistaa, ja onnistuinkin täyttämään suurimman osan listan vaatimuksista. Tärkein neuvo mielestäni oli: ”Yksinkertaista. Älä yritä luoda kerralla liian monimutkaista. Yksinkertainen on helppo ylläpitää ja toimii suuremmalla todennäköisyydellä.”. Tämä neuvo sivuaa tutkimusongelmaani ja ajan kuluessa osoittautuikin työni kulmakiveksi. Penttisen tarkistuslista on tarkasteltavissa kuvassa 16.

Tarkistuslista

Älä testaa käyttäjän kärsivällisyyttä

- Aseta sisältö heti tai helposti saataville. Älä piilotele sitä.
- Mieti sivuston latausaikoja. Pakkaa sisältöä jos se on mahdollista.
- Jatka-tästä-kotisivuilleni-sivut (introsivut) ovat turhia.

Yritä ennakoida virheitä

- Varmista että sivu on luettava ilman kuvia, tyyliiedostoja ja erillisiä skriptejä.
- Yritä rakentaa mahdollisemman monella selaimella ja selainalustalla toimiva sivu. Muista testata myös vanhemmilla selainversioilla.
- Yritä noudattaa jokaisen koodauskielen standardeja.
- Yksinkertaista. Älä yritä luoda kerralla liian monimutkaista. Yksinkertainen on helppo ylläpitää ja toimii suuremmalla todennäköisyydellä.
- Jaksota kirjoittamasi koodi, tee siitä mahdollisemman helposti ymmärrettävää ja kommentoi sitä. Käytä pieniä kirjaimia koodin kirjoittamiseen.

Ajattele enemmän käyttäjää kuin itseäsi

- Ole informatiivinen.
- Erottele navigaatiot selkeästi varsinaisesta sisällöstä.
- Tee tekstistä tarpeeksi isoa ja selvästi luettavaa. Katso, että teksti on kopioitavissa.
- Tee jokaiseen kuvaan alt-kuvaus.
- Huomioi turvarajat. Vältä turhien rullauspalkkien ilmestymistä vaakasuunnassa.
- Tee ensin sisältö, sitten vasta tyyli.
- Mieti missä järjestyksessä sisältö sivuilla tulee.
- Tarkista linkkien toimivuus ja tee sivujen välinen liikkuminen mahdollisemman helpoksi käyttäjälle.
- Älä pakota linkkejä aukeamaan uudessa ikkunassa. Vältä ponnahdusikkunoiden (popup) käyttöä.

Kuvio 16. Tarkistuslista (Penttinen 2008)

Sivustot on tehty XHTML- ja CSS- kielillä. Tyyliiedostoon on määritelty elementtien sijainnit ja koot. Yläosan (header) korkeus on 242 pikseliä ja leveys 100 prosenttia. Kuvassa 16 näkyy määrittelyt tyyliiedostosta.

```
#header
{
    height:242px;
    width:100%;
    background:url(ylaosa2.png) top left no-repeat transparent;
    position:relative;
}
```

Kuvio 17. Yläosan (header) määrittelyt tyyliiedostossa

Navigaatiossa on eniten ominaisuuksia. Korkeudeksi on määritelty 23 pikseliä ja leveys on 100 prosenttia. Myös navigaation fontti sekä taustavärit ja –kuva on määritelty tyyliiedostossa. Tyyliiedostoon on jouduttu kirjoittamaan paljon koodia, jotta navigaatiosta saataisiin halutunlainen. Kuitenkaan päivit-

täjän ei tarvitse tästä välittää, sillä jos hän haluaa esim. lisätä linkkejä navigaatioon, riittää, että tekee muutokset XHTML- tiedostoon.

```

#nav
{
    height:23px;
    padding:5px 10px;
    background-color:#45a6f1;
    position:relative;
}

#nav ul
{
    width:100%;
    font-family: Arial, verdana;
    font-size: 14px;
    margin:0;
    padding:0;
    list-style:none;
}

#nav ul li {
    display: block;
    position: relative;
    float: left;
}

#nav li ul
{
    display: none;
}

#nav ul li a
{
    display: block;
    text-decoration: none;
    color: #ffffff;
    border-top: 1px solid #ffffff;
    padding: 5px 21px 5px 20px;
    background-image:url('nappi.png');
    margin-left: 1px;
    white-space: nowrap;
}

#nav ul li a:hover
{
    background: #ffba00;
}

#nav li:hover ul
{
    display: block;
    position: absolute;
}

#nav li:hover li
{
    float: none;
    font-size: 11px;
}

#nav li:hover a
{
    background: #617f8a;
}

li:hover li a:hover { background: #95a9b1; }
#nav li
{
    display:inline;
    margin:0;
    padding:0;
}

```

Kuvio 18. Navigaation (nav) määrittelyt tyylitiedostossa

Sisällölle on määritelty leveydeksi 100 prosenttia ja korkeus määräytyy automaattisesti sisällön määrän mukaan.

```

#main
{
    padding:20px;
    width: 100%;
    position:relative;
    background-image: url(valiosa3.png);
    background-repeat:repeat-y;
}

```

Kuvio 19. Sisällön (main) määrittelyt tyylitiedostossa

Alaosan (footer) korkeudeksi on määritelty 153 pikseliä ja leveydeksi 100 prosenttia. Footer- elementtiin jouduttiin lisäämään clear- muotoilu. Sen avul-

la voidaan määrittää, saako halutun elementin vierellä olla kelluvia elementtejä vai ei. Jos kelluvia elementtejä ei sallita, clear- muotoilun alaisena oleva elementti ei nouse kelluvien elementtien yläpuolelle, vaan pysyy niiden alla. (Valkeinen 2010.)

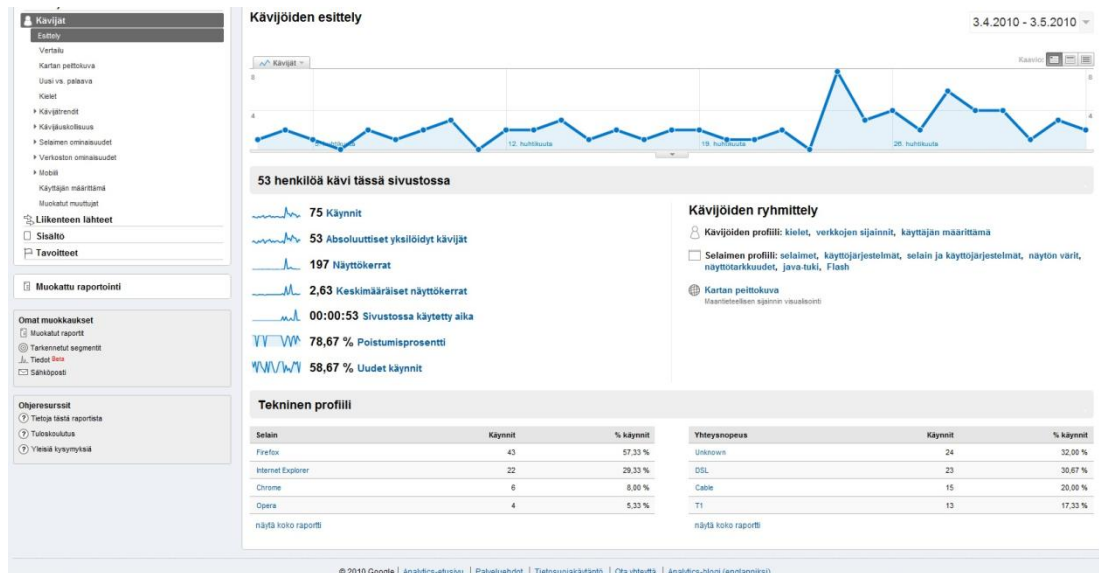
```
#footer
{
    clear:both;
    padding:5px 10px;
    height:153px;
    width:100%;
    background:url(alaosa3.png) top left no-repeat transparent;
    position:relative;
}
```

Kuvio 20. Alaosan (footer) määrittelyt tyylitiedostossa

Sivuston ulkoasu on kokonaisuudessaan määritelty tyylitiedostossa, lukuun ottamatta keskitystä <center>, jota on käytetty joissain teksteissä. Tyylitiedosto löytyy liitteestä 1.

5.4 Google Analytics

Lisäsin jokaiselle sivulle Google Analyticsin seurantakodin, jolloin toimeksiantajalla on mahdollista seurata liikennettä sivustolla. Google Analyticsia varten tein toimeksiantajalle henkilökohtaiset tunnukset palveluun. Google Analytics on verkkoanalyysiratkaisu, joka antaa tietoja verkkosivuston liikenteestä ja markkinoinnin tehokkuudesta.



Kuvio 21. Esimerkki Google Analyticsin sivusta

Google Analyticsin asentamiseen tarvitset henkilökohtaisen seurantakoodin, joka löytyy Google Analyticsin Profiilin asetukset – sivulta. Koodi lisätään sivun sisältöosan jälkeen välittömästi ennen kunkin seurattavan sivun </body> -koodia.

```
<script type="text/javascript">
var gaJsHost = (("https:" == document.location.protocol) ? "https://ssl." : "http://www.");
document.write(unescape("%3Cscript src='" + gaJsHost + "google-analytics.com/ga.js' type='text/javascript'%3E%3C/script%3E"));
</script>
<script type="text/javascript">
try {
var pageTracker = _gat._getTracker("UA-xxxxxx-x");
pageTracker._trackPageview();
} catch(err) {}
</script>
```

Kuvio 22. Google Analyticsin seurantakoodi

5.5 Testaus

Testausta varten siirsin sivut omalle henkilökohtaiselle sivutilalleni. Sen jälkeen valjastin kaikki tutut ja sukulaiset testaamaan sivustoja, jotta nähtäisiin niiden toimivuus eri selaimilla ja Internet- yhteyksillä. Itse testasin myös sivut useilla eri selaimilla. Käytössäni oli Firefox 3.6.3, Internet Explorer 8, Opera 10.51 ja Google Chrome 5.0.375.55.

Eri selaimet näyttävät CSS- muotoilut hieman eri tavoin, esimerkiksi tekstin koko yleensä vaihtelee selaimien välillä huolimatta tarkasta koon määrittelys-

tä. Selaimet voivat näyttää ja tulkita CSS- muotoiluja väärin tai vajaasti. Esimerkiksi Internet Explorer 6 ymmärtää periaatteessa korkeutta ja leveyttä koskevat muotoilut, mutta näyttää niiden yhteydessä marginaalit väärin. Tämän vuoksi on tärkeää testata www-sivujen toimivuutta eri selaimilla. Vaikka sivu toimisi loistavasti yhdellä selaimella, se ei välttämättä tarkoita, että sivu näkyisi myös vierailijoille oikein. (Valkeinen 2010.)

Yksinkertainen koodi osoittautui tämänkin asian suhteen hyödylliseksi, sillä mitä monimutkaisemmaksi muotoilut käyvät, sitä todennäköisemmäksi käyvät myös erot eri selaimien tavasta näyttää sivu (Haverinen 2010). Aiemmin ongelmallisin selain CSS- kielen suhteen on ollut Internet Explorer, mutta tähän on tullut parannuksia Internet Explorer 7:n jälkeen. Kuitenkin selaintestissä ilmeni ongelmia ainoastaan Internet Explorerin kanssa. Virheet olivat onneksi pieniä, sillä ainoastaan navigaation sijoittelu poikkesi pari pikseliä tarkoitetusta sijainnista. Olin kuitenkin varautunut tähän, minkä vuoksi olin alun alkaenkin suunnitellut navigaation taustalle samanvärisen palkin, jolloin Internet Explorerin virheet eivät ole helposti havaittavissa.

Kuten aiemmin mainitsin, eräs suurimpia hyötyjä CSS- ja XHTML- kielien yhteiskäytöstä on pienempi koodimäärä, joka paitsi helpottaa sivuston päivittämistä, vaikuttaa myös käytettävyyteen. Pienemmällä koodimäärällä nimittäin latausajatkin ovat nopeammat ja halusin kokeilla, miten tämä teoria toimii käytännössä. Hitain testauksessa käytetty Internet- yhteys oli max. 384 kbit/s ja nopein 100Mbit/5Mbit. Testi osoitti, että kyseinen teoria piti paikkansa, sillä sivusto toimi ongelmitta jopa hitaimmalla yhteydellä. Latausajat olivat lyhyet, joten käytettävyys ei kärsi ja näin ollen sivusto ei tule hitauden vuoksi menettämään kävijöitä.

6 PÄIVITETTÄVYYS

Sivuston päivitettävyyden tutkiminen olikin loppujen lopuksi hankalampaa kuin olin kuvitellut. www-sivujen käytettävyys ja visuaalisuus ovat yleensä etusijalla, joten päivitettävyyteen keskittyvää materiaalia on harmittavan vähän. XHTML- ja CSS- kieliin keskittyvä kirjallisuus opettaa kyllä paljon kyseisistä kielistä, mutta nekin hyvin harvoin sivuavat päivitettävyyttä. Yleensä jos yritykseltä tilataan www-sivut, tarjotaan samalla joko päivityspalvelu tai työkalut, jotka tekevät päivittämisestä helppoa.

Jos kotisivujen päivitystarve on vähäinen, on kannattavinta ostaa palvelu sivuston tekijältä. Yleensä yritykset tarjoavat palvelua esim. kuukausimaksulla ja päivitys hoituu vaivattomasti ottamalla yhteyttä palvelun tarjoajaan. Tässä tapauksessa kyseinen vaihtoehto on kuitenkin poissuljettu, sillä toimeksiantaja haluaa itse hoitaa sivuston päivittämisen.

Yksinkertaisin vaihtoehto olisi ollut asentaa palvelimelle julkaisujärjestelmä. Julkaisujärjestelmällä viitataan yleensä laajemman sisällönhallintajärjestelmän julkaisupainotteiseen osaan ja tähän liittyviin ominaisuuksiin. Monien järjestelmien keskeisimmät ominaisuudet ovat WYSIWYG- editori ja sen muotoilunappulat. Yleensä julkaisujärjestelmällä pyritään nopeuttamaan ja helpottamaan verkkopalveluiden sisällöntuotantoa ja erityisesti julkaisuprosessia. Julkaisujärjestelmän todellinen tarve syntyy kuitenkin vasta kun aktiivisia ylläpitäjiä on useita ja verkkopalveluiden sisältö muuttuu erittäin usein. (Tolvanen 2008.)

Julkaisujärjestelmistä olisin valinnut Joomla!:n, sillä se on avoimen lähdekoodin ohjelma, jolloin ohjelman voi ladata ilmaiseksi ja ohjelmistoa voi muokata ja hyödyntää vapaasti ilman mitään lisenssimaksuja. Joomla! asennetaan web- palvelimelle ja ohjelman ylläpitoliittymää käytetään internet- selaimella, eikä vaadi muita ohjelmia. Toimiakseen Joomla! vaatii PHP- ja MySQL- tuen palvelimelta. Peruskäytössä Joomla!:n hallitsemiseksi tarvitsee

vain osata käyttää tekstinkäsittelyohjelmaa. Joomla!-ssa voi myös helposti vaihtaa sivuston ulkoasua halutessaan. (Joomlaportal.fi 2007.)

Kaikin puolin julkaisujärjestelmä olisi siis ollut täydellinen vaihtoehto tällaisessa tilanteessa, mutta kuten aiemmin mainitsin, osoittautui tämä suunnitelma mahdottomaksi toteuttaa. Toimiakseen Joomla! pitäisi asentaa palvelimelle jo sivuston luomisvaiheessa, mikä ei tässä tilanteessa onnistunut, sillä toimeksiantajan nykyiselle palvelimelle minulla ei ole pääsyä. Palvelimen vaihto on edessä, mutta tässä vaiheessa siitä ei tietenkään ollut hyötyä.

Yksi vaihtoehto olisi ollut valita käyttöön jo aiemmin mainittu Macromedia Dreamweaver, mutta tämä ohjelma tosiaan on maksullinen, ja näin ollen ei käynyt. Aiemmat sivut oli toteutettu ja päivitetty Microsoft FrontPagella, eikä toimeksiantaja ollut tyytyväinen tähän vaihtoehtoon ja toivoikin, ettei sivuja varten tarvitse mitään erillistä ohjelmaa. Kaikkien rajoitteiden jälkeenärkevin vaihtoehto olikin tuottaa mahdollisimman pelkistettyä XHTML- ja CSS- koodia, jolloin päivittäminen onnistuu ilman ylimääräisiä ohjelmia ja hoituu tarvittaessa vain pelkän Notepadin avulla.

Ainoat ohjeistukset päivityksen helpottamiseksi näille kielille oli, että koodi tulisi pitää mahdollisimman yksinkertaisena. Lopputuloksena on saatu koodia, joka on selkeää ja johon toimeksiantajan tarvitsee tehdä muutoksia vain yhteen kohtaan halutessaan päivittää sivuja. Koodi pysyy yksinkertaisena ja selvänä etenkin sen vuoksi, että CSS- koodi on tehty ulkoiseen tyylitiedostoon. Tyylitiedosto on kuitenkin sellainen, ettei toimeksiantajan ole tarkoituskaan tehdä siihen muutoksia, sillä siinä määritellään jokaisen sivun ulkoasu. Tulevaisuudessa on hyvä, jos toimeksiantaja tekee ulkoasun muutokset täysin uuteen tyylitiedostoon, jolloin vältetään koodin puuroutumiselta ja epäloogisuudelta. Pyrin myös selventämään koodia kommentoinnin avulla, sillä koodia tarkastella toimeksiantaja saa sen avulla selvän kuvan koodin rakenteesta. Tähän mennessä päivitykset ovat yleensä olleet ajankohtaisten ta-

pahtumien ja aikataulujen muutoksia, joten toimeksiantajan tarvitsee periaatteessa osata käyttää vain paria tagia.

7 YHTEENVETO

Aiheen opinnäytetyölleni sain syksyllä 2009. Toimeksiantajan kanssa sovimme toteutusajankohdaksi kevään 2010, omien ja toimeksiantajan kiireiden vuoksi. Aihe kiinnosti minua erityisesti siksi, etten ole XHTML- tai CSS- kielillä aiemmin mitään itse toteuttanut ja oli inspiroivaa tehdä opinnäytetyö uusilla työvälineillä.

Uuden kielen opettelu toi mukanaan joitain hankaluuksia. Alkuperäiset luonnokset poikkeavat paljonkin nykyisestä, erityisesti koodipohjalla. Pari viikkoa ennen opinnäytetyön palautusta koko XHTML- koodi menikin kokonaan uusiksi perustavanlaatuisen virheen vuoksi. Siinä vaiheessa vasta opinkin ymmärtämään koodia kunnolla.

Eniten aikaa työssä meni ulkoasun suunnitteluun vähäisen ohjeistuksen vuoksi. Sivut ovat positiivinen parannus edellisiin. Päivitettävyyden oli avainasemassa, ja koenkin onnistuneeni siinä hyvin. Toimeksiantaja ymmärtää koodin rakenteen, mutta hänellä ei ole aiempaa kokemusta XHTML- kielen käytöstä. Tämän vuoksi pidänkin koulutuksen asian pohjalta sopivana ajankohtana. Tämän avulla pyrin varmistamaan, ettei toimeksiantajalla olisi tulevaisuudessakaan ongelmia päivitettävyyden kanssa. Kyseinen koulutus ei kuitenkaan sisälly opinnäytetyöhöni, vaan on oma erillinen projektinsa.

XHTML- ja CSS- kielten saralla tämä oli hyvin arvokas oppitunti, ja uskon tulevaisuudessakin käyttäväni näitä kieliä. Jos en ammatillisesti, niin ainakin omaksi ilokseni.

LÄHTEET

- ATPM 1998. WYSIWYG: Is it What You Want? Osoitteessa <http://www.atpm.com/4.12/page7.shtml> 1.12.1998
- Bonweb 2010. Logo – yrityksen käyntikortti maailmalle. Osoitteessa <http://www.bonweb.fi/blogi/?k=logo>. 25.1.2010
- Haverinen, Rauli 2010. Verkkopas. Osoitteessa <http://verkko-opas.fi/?id=97>. 5.2.2010
- Jaakkola, Tatu 2010. HTML-opas: Tyyliopohjat. Osoitteessa http://www.sivut.org/html/oppaat/tyyliopohjat.php#ulkoisessa_tiedostossa. 16.5.2010
- Joomlaportal.fi 2007. Mikä on Joomla!? Osoitteessa <http://www.joomlaportal.fi/content/view/93/39/> 17.11.2007
- Keränen, Vesa - Lamberg, Niko – Penttinen, Jukka 2003. Verkkopublikaisun hallinta. Docendo, Jyväskylä.
- Korpela, Jukka K. – Linjama, Tero 2004. XHTML- käsikirja. Docendo, Jyväskylä.
- Koulutus- ja konsultointipalvelu KK Mediat 2010. CSS opas. Osoitteessa <http://www.2kmediat.com/css/>. 16.5.2010
- MVnet 2010. Ulkoasu kotisivuille. Käytä valmiita ulkoasuja pohjana. Osoitteessa http://www.mvnet.fi/index.php?osio=Kotisivun_teko&sivu=Ulkoasu_kotisivuille. 16.5.2010
- Penttinen, Jani 2008. xHTML+CSS Survival Kit. Osoitteessa <http://survival.hollow13.net/>. 16.5.2010
- Taavetti, Riikka 2007. CSS:n ideasta ja mahdollisuuksista. Osoitteessa <http://www.cs.helsinki.fi/u/taavetti/>. 16.5.2010
- Tolvanen, Perttu 2008. Julkaisujärjestelmät Suomessa, markkinakatsaus 2008. Osoitteessa <http://vierityspalkki.fi/2008/03/31/julkaisujarjestelmat-suomessa-markkinakatsaus-2008/> 31.3.2008
- Valkeinen, Sonja 2010. CSS 2.1. Osoitteessa <http://www.menthal27.net/css/#alku>. 16.5.2010

LIITTEET

Liite 1: Tyylitiedosto

```
/****** tausta ja fontit *****/
```

```
body
```

```
{  
  
    height:100%;  
    background-image:url('tausta3.jpg');  
    background-repeat:repeat-x;  
    font: 84%/170% Arial, Tahoma, sans-serif;  
    color: #000b7a;  
  
}
```

```
#wrap
```

```
{  
  
    width:813px;  
    min-height:606px;  
    margin:0 auto;  
  
}
```

```
/****** yläkuva *****/
```

```
#header
```

```
{  
  
    height:242px;  
    width:100%;  
    background:url(ylaosa2.png) top left no-repeat transparent;  
    position:relative;  
  
}
```

```
/****** navigointi *****/
```

```
#nav
{
    height:23px;
    padding:5px 10px;
    background-color:#45a6f1;
    position:relative;
}

#nav ul
{
    width:100%;
    font-family: Arial, Verdana;
    font-size: 14px;
    margin:0;
    padding:0;
    list-style:none;
}

#nav ul li {
    display: block;
    position: relative;
    float: left;
}

#nav li ul
{
display: none;
}

#nav ul li a
{
display: block;
text-decoration: none;
```

```
    color: #ffffff;
    border-top: 1px solid #ffffff;
    padding: 5px 21px 5px 20px;
    background-image:url('nappi.png');
    margin-left: 1px;
    white-space: nowrap;
}

#nav ul li a:hover
{
    background: #ffba00;
}

#nav li:hover ul
{
    display: block;
    position: absolute;
}

#nav li:hover li
{
    float: none;
    font-size: 11px;
}

#nav li:hover a
{
    background: #617F8A;
}

li:hover li a:hover { background: #95A9B1; }
#nav li
{
    display:inline;
```

```
margin:0;
padding:0;
}

/***** sisältö *****/
#main
{
padding:20px;
width: 100%;
position:relative;
background-image: url(valiosa3.png);
background-repeat:repeat-y;
}

/***** alaosa *****/
#footer
{
clear:both;
padding:5px 10px;
height:153px;/*211*/
width:100%;
background:url(alaosa3.png) top left no-repeat transparent;
position:relative;
}
}
```