

RESURSSIENHALLINTAA

PHP-KALENTERISOVELLUS

LAHDEN AMMATTIKORKEAKOULU
Tietotekniikka
Opinnäytetyö
Kevät 2008
Mikko Vahter

Lahden ammattikorkeakoulu
Tietotekniikan koulutusohjelma

MIKKO VAHTER: Resurssienhallintaa – PHP kalenterisovellus

Tietotekniikan opinnäytetyö, 40 sivua, 14 liitesivua

Kevät 2008

TIIVISTELMÄ

Tässä työssä toteutetaan kalenterisovellus, jolla hallitaan töille tekijöitä. Sovellus toteutetaan PHP- ja AJAX-tekniikoilla. Sovellus tulee usean käyttäjän käyttöön, ja sitä käytetään useilta työasemilta verkkoympäristössä, mahdollisesti samanaikaisesti. Sovelluksen alustana toimii XAMPP-palvelinpaketti Windows Server 2003-palvelimessa. Windows toimialueen ohjauskone tunnistaa kalenterin käyttäjät ja tietokoneet sekä hoitaa kalenteritietojen varmistuksen.

Kalenteria käytetään Hollolan Sähköautomaatiikka Oy:n työnvarausjärjestelmänä, ja kalenteri on kaikkien yrityksen työntekijöiden käytettävissä. Sovellus toteutetaan verkkoympäristöön, jotta sovellusta voidaan käyttää kotoa tai työpaikalta käsin. Asiakkaan kysellessä tuotettaan kalenterista nähdään samantien, koska työ on valmistumassa tai koska se on mahdollista valmistaa. Myös lomat ja tapaamiset voidaan merkitä kalenteriin.

Kalenterisovelluksen pohjana käytetään valmiita kirjastoja ja avoimen lähdekoodin sovelluksia. Sovellusta käytetään www-selaimella, mutta käyttöliittymä on mahdollisimman työpöytäsovellusmainen. Käyttöliittymästä pyritään tekemään myös mahdollisimman helppokäyttöinen. Kalenterissa on kaikille käyttäjille oma kalenteriluokkansa, johon merkintöjä voidaan luoda. Kalenterimerkintöjä voidaan myös siirtää tai venyttää sekä muuttaa merkinnän kalenteriluokkaa. Merkintöjä voidaan myös merkitä tehdyksi yliviiwaamalla ne. Eri henkilöiden kalenterimerkinnät näytetään eri väreillä, joten omat merkinnät on helppo erottaa muista. Omat merkinnät voi myös helposti tulostaa paperille, joten tehtyjen töiden seuraaminen helpottuu.

Avainsanat: Kalenteri, AJAX, PHP

Lahti University of Applied Sciences
Faculty of Technology

MIKKO VAHTER: Resource management with PHP application

Bachelor's Thesis in Information Technology, 40 pages, 14 appendices

Spring 2008

ABSTRACT

The aim was to develop a user-friendly time allocation system. The work was commissioned by Hollolan Sähköautomaatiikka Oy. The application helps company to manage its work resources.

A calendar is used in network environment. The application is implemented with PHP and AJAX technology. It is used simultaneously by multiple users and it runs on a Windows server, with a XAMPP server package installed. The server recognizes calendar users and computers with Windows active directory technology.

The calendar is used as a time allocation system for workers and it has to be available to all employees. The calendar application is based on multiple libraries and open source applications. Despite being used with a web browser, the calendar looks and feels like a desktop application. In this application, the calendar events can be moved or stretched to multiple days. Events for different employees are shown with different colors. It is easy to mark holidays, jobs and meetings to this calendar.

The calendar was developed successfully and users have started to use it. The feedback has been positive.

Keywords: Calendar, AJAX, PHP

SISÄLTÖ

LYHENNELUETTELO

1	JOHDANTO	1
2	TEKNIIKAT, KIELET JA KIRJASTOT	2
2.1	Yleistä	2
2.2	Palvelin ja asiakassovellukset	2
2.3	PHP	3
2.4	AJAX	5
2.5	PHPiCalendar	6
2.6	XAMPP	8
2.7	Javascript	9
3	WEB-DOKUMENTIT	11
3.1	Staattinen ja dynaaminen web-dokumentti	11
3.2	HTML	13
3.3	CSS	13
3.4	HTTP-protokolla	15
3.5	XML	16
3.6	DOM JA SAX	18
3.7	RSS	23
4	TOTEUTUS	24
4.1	Monket-Calendar	24
4.2	Työn toteutus	26
5	YHTEENVETO	35
	LÄHTEET	36

LYHENNELUETTELO

AJAX (Asynchronous Javascript and XML) sovellukset muokkaavat www-sivuja dynaamisesti, ilman käyttäjän navigointia

PHP (PHP Hypertext Preprocessor) on palvelimella suoritettava, tulkittava komentosarjakieli

HTTP (HyperText Transfer Protocol) on Web-palvelimen ja asiakkaan välisessä tiedonsiirrossa käytettävä yhteyskäytäntö

HTML (HyperText Markup Language) on World Wide Webin sivunkuvauskieli

XHTML (eXtensible HyperText Markup Language) XHTML koostuu samoista asioista kuin HTML, mutta sen kielioppia on muutettu XML-sääntöjen mukaiseksi

XML (eXtensible Markup Language) on merkintäkieli, jolla tiedon merkitys on kuvattavissa tiedon sekaan

CSS (Cascading Style Sheets) tarkoittaa putousmallista tyylimallia

DOM (Document Object Model)-mallia käytetään sivujen esittämiseen ja käsittelyyn

SAX (Simple API for XML) on yksinkertainen rajapinta XML:lle

APACHE (A Patchy Server) on eniten käytetty web-palvelinohjelmisto

XAMPP (X Apache MySQL PHP Perl) on helppokäyttöinen ja ilmainen APACHE asennuspaketti

RSS (Really Simple Syndication tai Rich Site Summary) eli uutissyöte on XML-kielen laji

DTD (Document Type Definition) on dokumentin rakennekuvaus

PERL (Practical Extraction and Report Language) on skriptimäinen ohjelmointikieli

1 JOHDANTO

Tässä työssä toteutetaan kalenterisovellus verkkoympäristöön. Sovelluksella hallitaan töitä sekä varataan töille tekijöitä. Keskitetty resurssienhallinta säästää yrityksen aikaa ja rahaa, asiakkaiden tiedusteluihin voidaan vastata helposti, koska työn arvioitu valmistumispäivä nähdään samantien kalenterista. Myös lomien ja poissaolojen merkitseminen on helppoa.

Sovellus toteutetaan PHP- ja AJAX-tekniikoilla. Sovellus tulee usean käyttäjän käyttöön, ja sitä käytetään useilta työasemilta verkkoympäristössä, mahdollisesti samanaikaisesti. Sovelluksen alustana toimii XAMPP-palvelinpaketti Windows palvelimessa. Windows toimialueen ohjauskone tunnistaa käyttäjät ja tietokoneet, eikä erillistä käyttäjätunnistusta tarvita.

Kalenteria tullaan käyttämään Hollolan Sähköautomaatiikka Oy:n työnvarausjärjestelmänä ja kalenterin tulee olla kaikkien yrityksen työntekijöiden käytettävissä. Hollolan Sähköautomaatiikka Oy on varavoimalaitteisiin ja niihin liittyvään teknologiaan erikoistunut, noin 10 henkilöä työllistävä yritys. Kalenterisovelluksen pohjana käytetään valmiita kirjastoja ja avoimen lähdekoodin sovelluksia. Kalenteria käytetään www-selaimella, ja siitä pyritään tekemään mahdollisimman helppokäyttöinen. AJAX-tekniikan avulla kalenterisovelluksesta voidaan tehdä työpöytäsovellusta muistuttava.

2 TEKNIKAT, KIELET JA KIRJASTOT

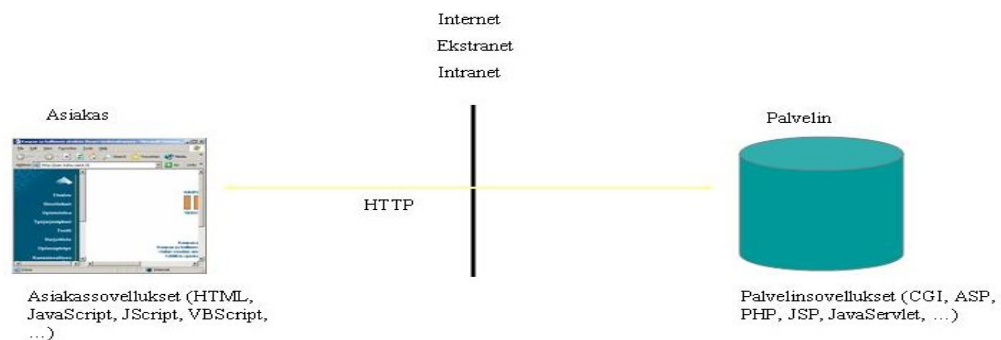
2.1 Yleistä

Kalenterisovelluksen toteutus verkkoympäristöön tarjoaa käyttäjille mahdollisuuden käyttää kalenteria mistä tahansa – edellyttäen että internetyhteys on käytettävissä. Verkkoympäristön etuna on myös se, että työasemille ei tarvitse asentaa mitään, vaan kalenteria voi käyttää web-selaimella – joka löytyy nykyaikaista työasemista valmiina. Sovelluksesta tehdään mahdollisimman paljon perinteistä työpöytäsovellusta muistuttava, käyttäjäystävällinen sovellus.

2.2 Palvelin ja asiakassovellukset

Web-sovellukset on toteutettu palvelin tai asiakastekniikalla tai molemmilla. Asiakastekniikalla toteutettu sovellus suoritetaan käyttäjän selaimessa ja palvelinsovellus taas web-palvelimella.

Palvelinsovellusta käytettäessä käyttäjä hakee sivun ja tällöin pyyntö välittyy web-palvelimelle, jossa esimerkiksi PHP-sovellus suoritetaan ja tulos välitetään takaisin käyttäjän selaimelle. Kuviossa 1 on kuvattu eri asiakas- ja palvelinsovelluksia.



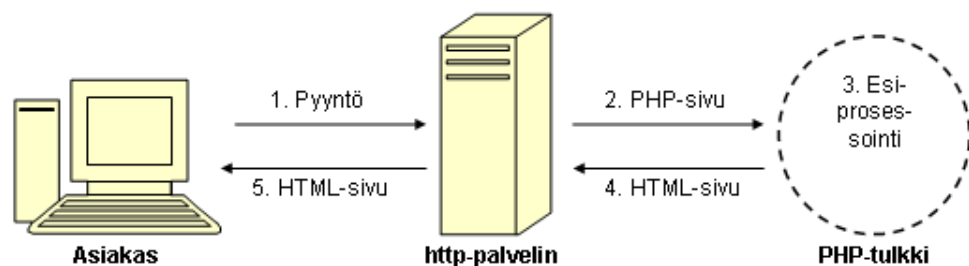
KUVIO 1. Asiakas ja palvelinsovellukset (Oulun seudun ammattikorkeakoulu 2003-2006)

Palvelinsovellukset vaativat palvelimelle erilaisia palvelinohjelmistoja.

Asiakassovellusten toteuttamisessa tulee ottaa huomioon, että käyttäjän selaimen on tuettava käytettyä toteutustekniikkaa. Toteutustekniikkana voi olla skriptikieli, sovelma tai jokin plug-in tyyppinen ohjelma (Oulun seudun ammattikorkeakoulu 2003-2006).

2.3 PHP

PHP (PHP Hypertext Preprocessor) on palvelimella suoritettava, tulkettava komentosarjakieli. PHP muistuttaa syntaksiltaan C-, Java- ja Perl-kieliä. PHP-sivut käsitellään www-palvelimella, ennen kuin sivu ladataan käyttäjän selaimelle (kuvio 3). PHP:tä käytetään usein toiminnallisten www-sivujen toteuttamiseen. Koska PHP suoritetaan palvelimella, se ei tarvitse selaintukea, ja skriptillä on myös pääsy erilaisiin palvelimen tietokantoihin ja tiedostoihin.



KUVIO 3. PHP:n toiminta (Kollanus 2005)

PHP-kieli on helppo omaksua, jos on ohjelmoinut aikaisemmin jollain muulla kielellä. PHP on hyvin dokumentoitu, ja sitä käytetään laajalti ilmaisissa sekä kaupallisissa sovelluksissa. PHP:n kovin kilpailija on Microsoftin ASP (Active Server Pages) (2kmediat 2008).

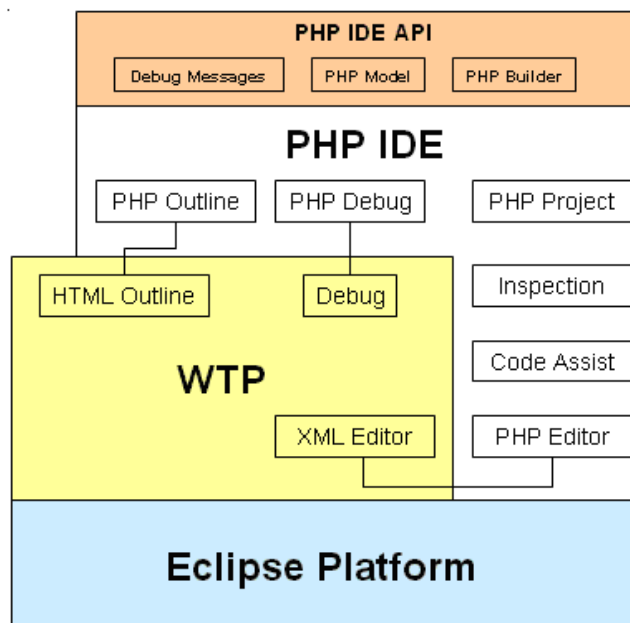
PHP on suunniteltu erityisesti web-sovelluskehitykseen. Se on avoin ohjelmisto, jonka lähdekoodi on saatavilla ilmaiseksi. PHP ohjelmien suorittamiseen tarvitaan siis Web-palvelinohjelmisto ja PHP-tulkki.

Yleisimmin PHP:tä käytetään Apache-palvelimen kanssa, mutta yhtä hyvin sitä voidaan käyttää vaikka Microsoftin IIS palvelimessa (Rantala 2002, 12).

PHP-kieli on helposti opittava ja selaimelta saatava data on suoraan käytettävissä. PHP on myös heikosti tyyhitetty, joten esim. muuttujien tyyppiä ei tarvitse määrittellä etukäteen. Syntaksi on tuttu C-kielestä.

Tanskalainen Rasmus Lerdorf on julkaissut PHP-kielen ensimmäinen version vuonna 1995. Jo 1998 julkaistu PHP 3 sisälsi jonkinlaisen tuen olio-ohjelmoinnille, mutta vasta vuonna 2004 julkaistu PHP5 sisältää lähes täydellisen tuen olio-ohjelmoinnille.

PHP-koodi kirjoitetaan yleensä tekstieditoreilla tai HTML-editoreilla, varsinaisia PHP integroituja kehitysympäristöjä (IDE, Integrated Development Environment) on myös muutamia (Rantala 2002, 14-15). Kuviossa neljä on esitetty Eclipse PHP-kehitysympäristön rakenne. Kehitysympäristön tärkeimpiä komponentteja ovat WTP (Web Tools Platform)-komponentit. WTP yhdistää asiakas- ja palvelinohjelmoinnin samaan kehitysympäristöön.



KUVIO 4. Eclipse project (Eclipse Foundation 2008)

2.4 AJAX

Ajax –sovellukset (Asynchronous Javascript and XML, XML on formaatti jota käytetään asiakkaan ja palvelimen välisessä tiedonsiirrossa) muokkaavat www-sivuja dynaamisesti, ilman käyttäjän navigointia. Ajaxin avulla www-sovellukset voidaan muuttaa muistuttamaan työpöytäsovelluksia. Ajax voi tarkistaa esim. lomakkeen tiedot samantien sekä kutsua PHP-skriptiä. Ajaxin avulla voidaan myös lomakkeen sisältöä voidaan muuttaa dynaamisesti käyttäjän syötteiden mukaan.

Ajax:issa XHTML- ja CSS-kieltä käytetään rakenteen ja esitystavan luomiseen, DOM-mallia (Document Object Model) käytetään sivujen esittämiseen ja käsittelyyn. AJAX-sovellus luo palvelimen ja käyttäjän väliin välittäjän, joka tuottaa tietoa käyttöliittymälle. Jos käyttäjän toiminto ei vaadi palvelinyhteyttä, Ajax-kone vastaa pyyntöön heti, muussa tapauksessa selain ei jää odottamaan palvelimen vastausta, vaan käyttäjä voi jatkaa sivun käyttöä, ja sivu päivitetään vastauksen saapuessa. Näin sovelluksista tulee toiminnoiltaan lähes yhtä nopeita kuin työpöytäsovellukset (Smith & Negrino 2007, 359-360).

Kuviossa kaksi esitetään kuinka, Ajax-sovellus tuottaa käyttöliittymälle tietoa käyttäjän syötteisiin perustuen. Quickstart.js Javascript tiedosto kutsuu palvelimella suoritettavaa quickstart.php-tiedostoa. Tiedoston suoritettuaan palvelin palauttaa tuloksen ja web-sivu päivitetään.

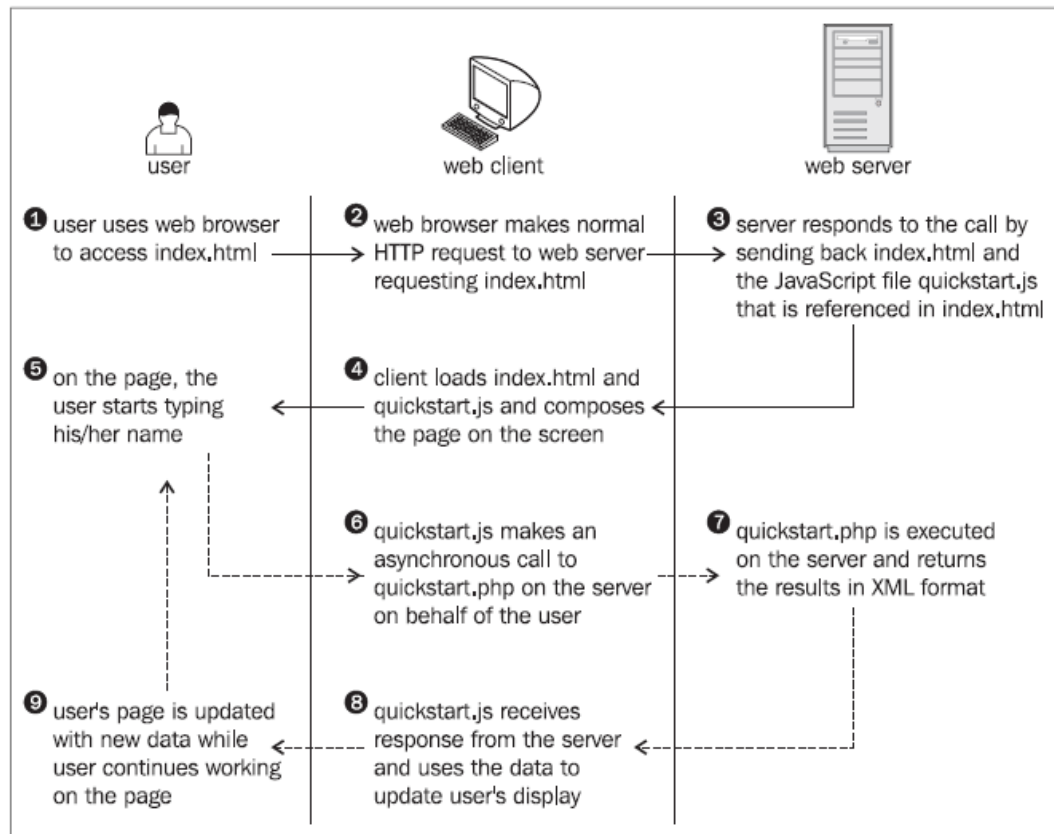


Figure 1.7: The Diagram Explaining the Inner Works of Your Quickstart Application

KUVIO 2. Ajax sekvenssikaavio (Darie)

2.5 PHPiCalendar

PHPiCalendar on PHP:hen pohjautuva iCal-tiedostoja käsittelevä selaimella käytettävä sovellus. iCal on alunperin Applen MAC OS:lle kehittämä kalenterisovellus. Microsoft Outlook ohjelmisto on osittain iCal-yhteensopiva. iCalendar on standardi kalenterien tiedonkäsittelylle.

PHPiCalendar on vain näkymä, jonka avulla iCal kalentereita voi tarkastella selaimella. IETF (Internet Engineering Task Force) on määritellyt iCal-tiedostot RFC 2445 (Request For Comments) dokumentissa (IETF, 1998).

iCal tiedosto tulee alkaa "BEGIN:VCALENDAR" ja päättyä "END:VCALENDAR" tekstillä. Näiden rivien välissä ovat kalenterimerkinnot.

Merkinnät alkavat ”BEGIN:VEVENT” ja päättyvät ”END:VEVENT” tekstillä (Google 2008). Kuviossa 5 on esitetty iCal-tiedoston rakenne.

```
BEGIN:VCALENDAR
(Otsikko tiedot tulevat tähän)
BEGIN:VEVENT
(Yksittäisen merkinnän tiedot)
END:VEVENT
BEGIN:VEVENT
(Toisen merkinnän tiedot)
END:VEVENT
END: VCALENDAR
```

KUVIO 5. iCal tiedoston rakenne (Google 2008)

iCal tiedosto on tuettuna monissa kalenterisovelluksissa, kuviossa 6 on esitetty Plazes kalenterisovelluksen käyttöliittymä mobiililaitteen näytöllä. iCal-tiedostoja voidaan synkronisoida moniin mobiililaitteisiin suoraan työasemalta erilaisten apuohjelmien avulla.



KUVIO 6. iCal tiedostoa käsittelevä kalenteri mobiililaitteessa (Plazes 2007)

2.6 XAMPP

XAMPP on helppokäyttöinen ja ilmainen APACHE (A Patchy Server) asennuspaketti, joka on saatavilla useille eri käyttöjärjestelmille. XAMPP sisältää apachen lisäksi esim. MySQL tietokantapalvelimen, PHP-tuen, Perlin ja OPENSSEL:n. Koska XAMPP ei ole tarkoitettu varsinaiseen tuotantokäyttöön vaan kehitysympäristöksi, niin oletuksena XAMPP-paketti on konfiguroitu tietoturva-mielessä avoimeksi, eikä se sovellu tuotantokäyttöön ennen asetusten konfigurointia (Apachefriends, 2002-2008).

XAMPP:ista on kehitetty erikseen ns. siirrettävä (portable) versio, jota voi käyttää USB-muistilta tai muulta siirrettävältä asemalta. XAMPP Lite on kevennetty versio, jossa on vain useimmin tarvittavat palvelut, eikä siihen ole myöskään saatavilla päivityspaketteja (Sanoma Magazines Finland, 2007).

APACHE on eniten käytetty web-palvelinohjelmisto, sitä käyttää noin 60 prosenttia web-palvelimista. APACHE kehitettiin vuonna 1995 HTTPd web-palvelinohjelmiston pohjalta, ja se on saatavilla useimpiin käyttöjärjestelmiin. APACHEn suosio perustuu sen ilmaisuuteen, luotettavuuteen, nopeuteen ja avoimeen lähdekoodiin (Ek & Eriksson 2001, 43).

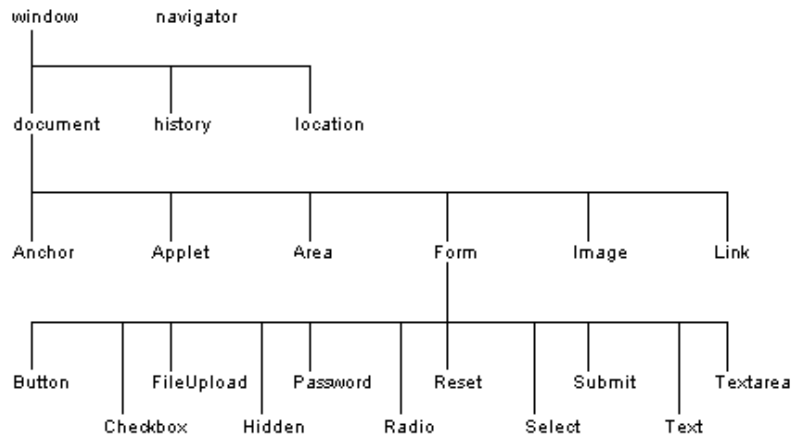
2.7 Javascript

Javascript on ohjelmointi- tai skriptikieli, jolla ei nimestään huolimatta ole mitään tekemistä Javan kanssa (Javascript tehokas hallinta, s.3). Javascriptiä käytetään toteuttamaan toiminnallisuutta www-sivustolla. Kielen perusteet on helppo oppia, jos hallitsee ohjelmoinnin perusteet. Javascript on suunniteltu yleiskäyttöiseksi skriptikieleksi, joka yleensä suoritetaan selaimessa, mutta se on mahdollista sulauttaa myös esim. web-palvelimiin. Javascriptin (alkuperäinen nimi LiveScript) kehitti alun perin Netscape, mutta Microsoft on tehnyt myös kielestä oman versionsa Jscriptin. JScript on pitkälti yhteensopiva Javascriptin kanssa, mutta molemmissa kielissä on omat erikoisominaisuutensa. EcmaScript on JavaScript-kielen standardisoitu versio, ja se on suunniteltu JavaScript- ja Jscript-kielien pohjalta.

Javascript-kieltä voi kirjoittaa haluamallaan tekstieditorilla tai HTML-editorilla. Useimmat HTML-editorit osaavat tuottaa ohjelmoijalle valmista koodia (Peltomäki Javascript Toolkit, 1-9).

Javascript-kielen toteutus perustuu kääntämisen sijaan tulkintaan. Javascript voidaan liittää HTML-dokumenttiin useilla tavoilla; Javascript koodi voi olla erillisessä .js tiedostossa, joka sisällytetään HTML-dokumenttiin, tapahtumamääritteet voi liittää suoraan HTML elementteihin, tai Javascript voi olla upotettuna HTML-dokumenttiin (Korpela, 2007).

JavaScript muodostaa kaikista www-dokumentin elementeistä ja ominaisuuksista hierarkian. Elementtejä ja ominaisuuksia voidaan käsitellä hierarkiaan kuuluvien olioiden avulla (KUVIO 7). Olioita kutsutaan selainolioiksi tai navigaattorolioiksi.



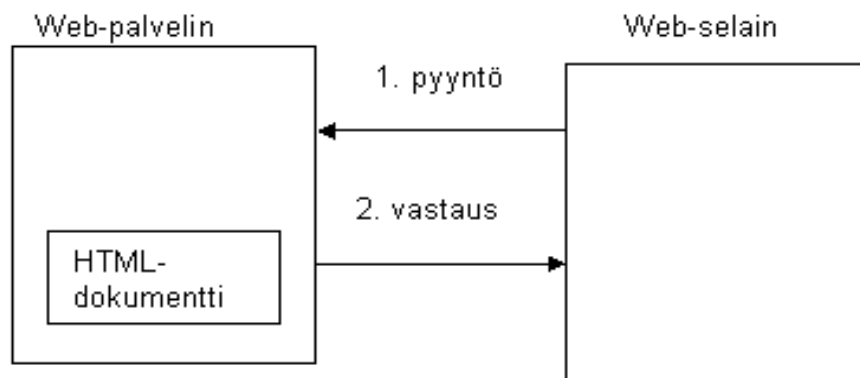
KUVIO 7. JavaScript hierarkia (Netscape 1996)

2 WEB-DOKUMENTIT

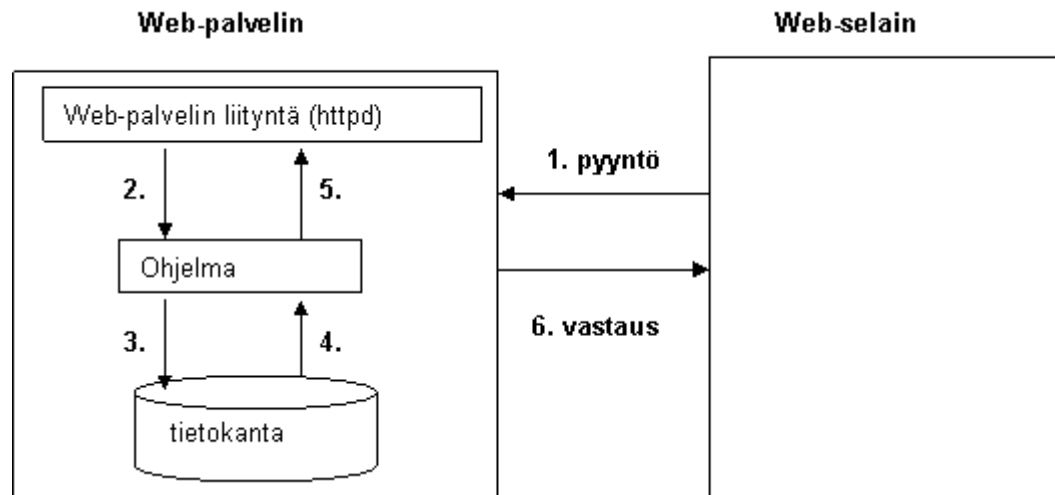
3.1 Staattinen ja dynaaminen web-dokumentti

Perinteisesti Web-dokumentit ovat olleet staattisia (kuvio 8). Tällöin niiden muokkaaminen vaatii Web-sivuston tekijän toimia. Staattisten dokumenttien muuttaminen vaatii käytännössä sivun uudelleen ohjelmointia sekä siirtoa Web-palvelimelle ja on sen vuoksi työlästä ja vaikeata. Sivuston käyttäjän ainoa mahdollisuus vaikuttaa esitettyyn tietoon on hyperlinkkien käyttäminen.

Dynaamiset Web-dokumentit sisältävät muuttuvaa tietoa. Tietoa voidaan hakea esimerkiksi tietokannasta käyttäjän syötteisiin perustuen. Usein dynaamisia Web-sivustoja käyttävät esimerkiksi erilaiset verkkokaupat. Yleensä Web-sivustot sisältävät sekä dynaamisia että staattisia dokumentteja. Dynaamiset dokumentit luodaan automaattisesti osittain tai kokonaan, ja tällöin puhutaan web-sovelluksesta (Rantala 2002, 5-6). Kuviossa 9 on esitetty dynaaminen web-dokumentti.

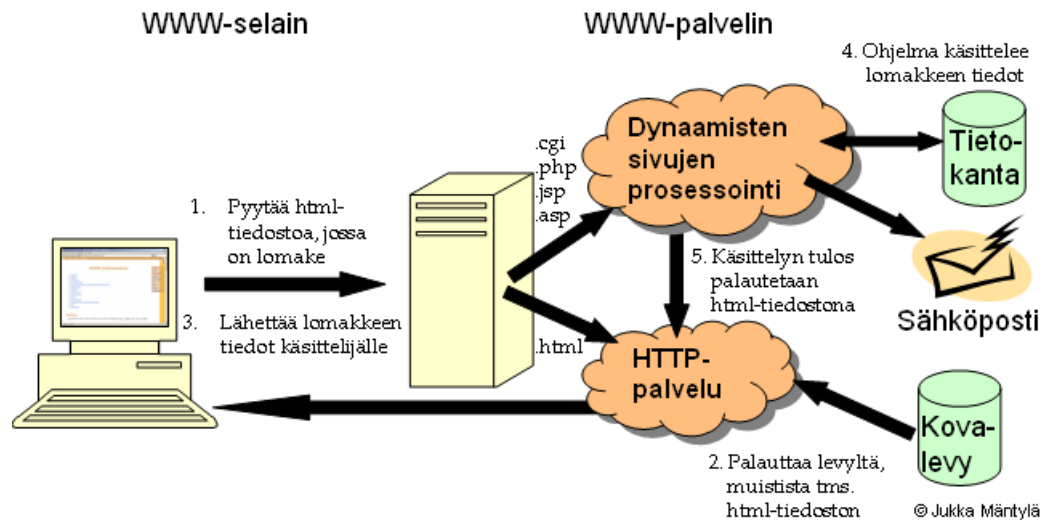


KUVIO 8. Staattinen web-dokumentti (Oulun seudun ammattiopisto 2006)



KUVIO 9. Dynaaminen Web-dokumentti (Oulun seudun ammattiopisto 2006)

Kuviossa 10 on kuvattuna dynaaminen tapahtuma kokonaisuudessaan: 1.) web-selain lähettää pyynnön palvelimelle, 2.) palvelin palauttaa html-tiedoston, 3.) selain lähettää lomakkeen tiedot käsittelijälle, 4.) ohjelma käsittelee lomakkeen tiedot, 5.) käsittelyn tulos palautetaan web-selaimelle html-tiedostona.



KUVIO 10. Dynaaminen tapahtuma (Jyväskylän avoin yliopisto 2006)

3.2 HTML

HTML (Hyper Text Markup Language) on merkintäkieli, jota käytetään www-dokumenttien luomiseen. Dokumentti sisältää varsinaisen tekstisisällön lisäksi ohjeita elementtien rakenteen ja ulkoasun määrittelemiseksi. HTML-dokumentti voi tekstin lisäksi sisältää linkkejä sekä viittauksia kuviin ja muihin mediatiedostoihin. HTML-dokumentit ovat päätteeltään .html tai .htm päätteisiä. Dokumentit voi kirjoittaa millä tahansa ASCII-tekstieditorilla, mutta usein käytetään erillisiä HTML-editoreita, jotka osaavat luoda valmista HTML-koodia.

Tiedosto, jonka nimi on index.html, on kunkin hakemiston oletustiedosto, jonka palvelinohjelmisto hakee automaattisesti, jos pyynnössä ei ole nimetty tiedostoa (Lehtonen, 2001).

HTML-kielestä pidetään sen helppokäyttöisyyden vuoksi: kielen oppiminen on nopeaa. Se ei kuitenkaan ole kovinkaan joustava kieli, ja sen vuoksi sen lisänä käytetään erilaisia skriptikieliä (Peltomäki 1998, 126).

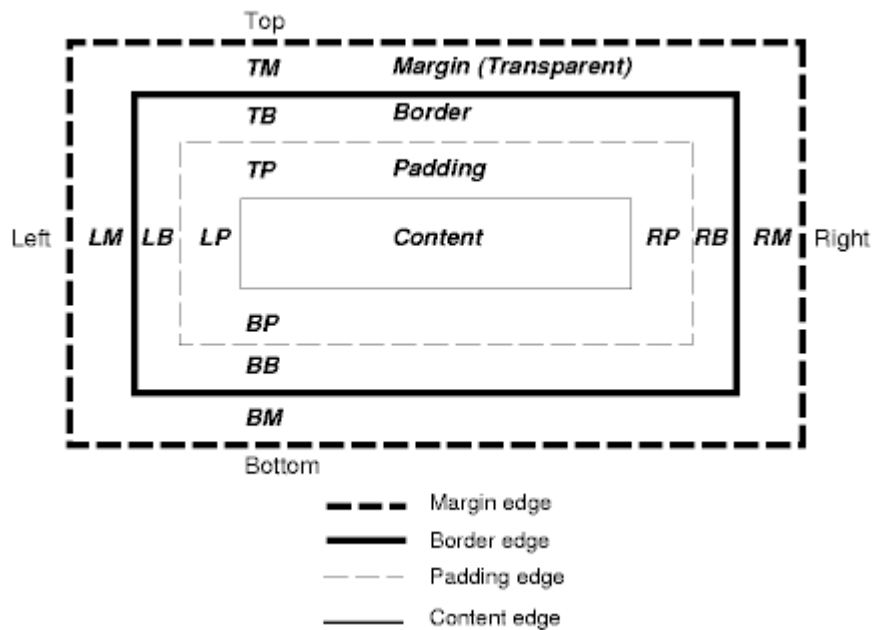
HTML:ää kutsutaan usein kieleksi, mutta tämä nimitys on harhaanjohtava, koska se ei ole ohjelmointikieli vaan rakenteinen dokumenttimuoto. HTML mukautuu hyvin erilaisiin esitystilanteisiin. Eri selaimet saattavat esittää HTML-dokumentin hyvinkin eritavoilla (Korpela, 2007).

3.3 CSS

CSS (Cascading Style Sheets) tarkoittaa putousmallista tyylimallia, eli tyylien määrittelyssä otetaan huomioon aikaisemmat tyylimäärittelyt. CSS-tyylejä tuetaan nykyään jo lähes kaikissa selaimissa. Tyyli määrittelee, miltä jokin HTML (Hypertext Markup Language) elementti näyttää dokumentissa. CSS-syntaksi poikkeaa hieman HTML-kielen syntaksista – esim. ominaisuudet erotetaan toisistaan puolipisteellä (Peltomäki 2001, 204).

Tärkeimpiin CSS:n etuihin kuuluu se, että päästään eroon html muotoilutageista ja sivun ulkoasu voidaan määrittellä paljon tarkemmin. CSS tukee suhteellisia sekä absoluuttisia mittoja. Suhteellisten mittojen koko riippuu jonkun tietyn elementin koosta, ja absoluuttiset mitat taas ovat kiinteitä (Smith & Negrino 2007, 268).

Kuviossa 11 on kuvattu CSS:n laatikkomalli (box-model). Laatikkomallin avulla kuvataan elementin ja dokumentin ulkoasua sekä hallitaan niiden piirteitä (2kmediat, 2000-2008). Laatikkomallin osia ovat: 1.) margin eli marginaalit joka on elementin ympärille jätettävä tyhjä tila, 2.) border eli reunukset tai reunaviivat, 3.) padding eli täyte on elementin sisällön ja reunan välinen tyhjä tila sekä 4.) content eli sisältö (Vanhala-Nurmi 2005).



KUVIO 11. CSS laatikkomalli (W3C 1998)

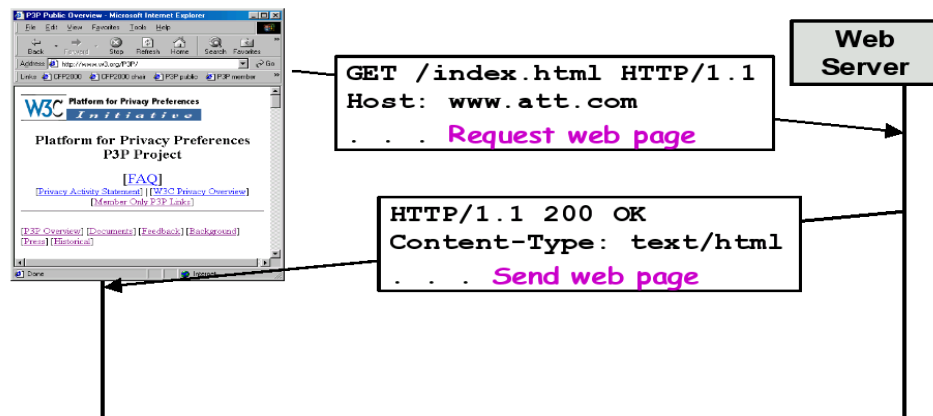
CSS-tyylimäärittely voidaan liittää HTML- tai XHTML-dokumenttiin useilla tavoilla.

Liittämistapoja ovat:

- rivinsisäinen tyylimäärittely, jossa tyylisääntö liitetään yksittäiseen elementtiin style-attribuutilla.
- tuotu tyylimäärittely, jossa ulkoinen tyylimäärittely tuodaan osaksi tyyliä @import-säännöllä.
- ulkoinen tyylimäärittely, jossa ulkoinen tyylitiedosto linkitetään dokumenttiin link-elementillä.
- upotettu tyylimäärittely, jossa tyylimäärittely upotetaan osaksi dokumenttia style-elementillä (2kmediat, 2000-2008).

3.4 HTTP-protokolla

HTTP (HyperText Transfer Protocol) on Web-palvelimen ja asiakkaan välisessä tiedonsiirrossa käytettävä yhteyskäytäntö. HTTP-protokollalla voidaan siirtää monenlaista dataa. HTTP-siirtotapahtuma rakentuu kahdesta eri HTTP-viestistä: asiakaspyynnöstä sekä palvelimen vastauksesta, siirtotapahtuma on esitetty kuviossa 12. HTTP-siirtotapahtuma alkaa, kun palvelin saa asiakaspyynnön, tämän jälkeen palvelin käsittelee pyynnön ja palauttaa asiakkaalle datan (Rantala 2002, 110). HTTP-siirtotapahtumaan kuuluu myös TCP-yhteyden (Transmission Control Protocol) muodostus sekä sulkeminen palvelimen ja asiakkaan välissä. Selaimen liikennöinti verkkoon tapahtuu socket-liittymän avulla. Socket-liittymä huolehtii verkkokortin ajurien, TCP/IP protokollapinon sekä pakettiajureiden kanssa TCP/IP-kehysten pakkaamisesta. HTTP-pyyntön mukana liikkuu myös otsikkokenttiä, joissa on tietoa HTTP versiosta sekä asiakasohjelmistosta (Peltomäki 1998, 20). Asiakkaan ja palvelimen välinen liikennöinti tapahtuu merkkimuotoisilla komennoilla sekä tilakoodeilla. HTTP 1.0 versiossa yhteys avattiin ja suljettiin jokaisen linkitetyn dokumentin yhteydessä, mutta HTTP 1.1 versiossa yhteyttä pidetään auki muutama sekunti ja yhteys suljetaan, jos palvelupyynnöt ei tule tänä aikana. Lähes kaikissa WWW-palvelimissa käytetään HTTP 1.1 versiota.



KUVIO 12. HTTP-siirtotapahtuma (World Wide Web Consortium 2002)

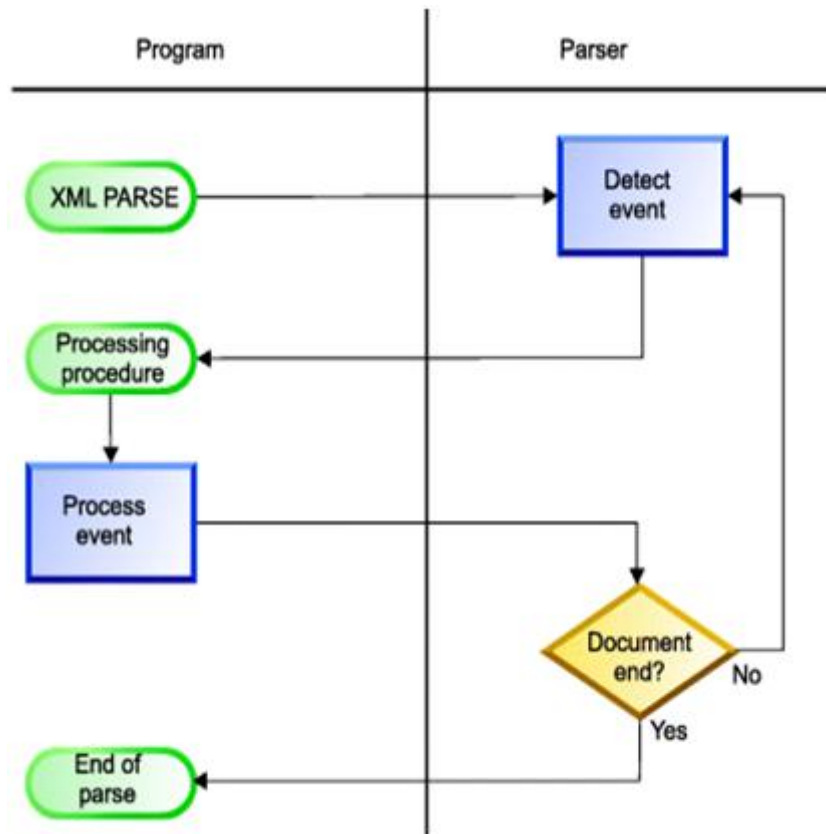
Koska HTTP on tilaton protokolla, niin peräkkäisistä siirtotapahtumista ei pystytä pitämään tilatietoa. Tilatiedon puutteesta johtuen jokainen siirtotapahtuma on aloitettava alusta. Web sovelluksen toiminnallisuus rakentuu peräkkäisistä HTTP-siirtotapahtumista (Rantala 2002, 110).

3.5 XML

XML (Extensible Markup Language) on tarkoitettu rakenteisen tiedon esittämiseen. XML-kielessä voidaan itse määrittää sovelluksessa, mitä jokin tagi tarkoittaa. XML ei ole ohjelmointikieli, vaan se on tekstiformaatti, joka käyttää tageja datan rajaamiseen. Eräs tärkeä XML-sovellus on XHTML – HTML:n seuraaja. XHTML koostuu samoista asioista kuin HTML, mutta sen kielioppia on muutettu XML-sääntöjen mukaiseksi (W3C Suomen toimisto, 1999-2003).

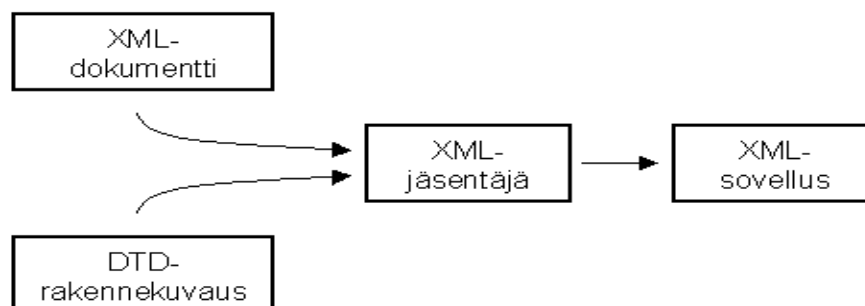
XML on käytetyin tekniikka sähköisten dokumenttien standardoinnissa. XML-kiehellä kirjoitettu dokumentti voidaan näyttää uudemmissa selaimissa, kuten HTML dokumentti, mutta tällöin on tehtävä tyylisivut.

Kuviossa 13 on esitetty SAX-tyyppinen, tapahtumapohjainen XML-dokumentin parseri eli jäsentäjä.



KUVIO 13. XML-Parser (IBM 1992-2006)

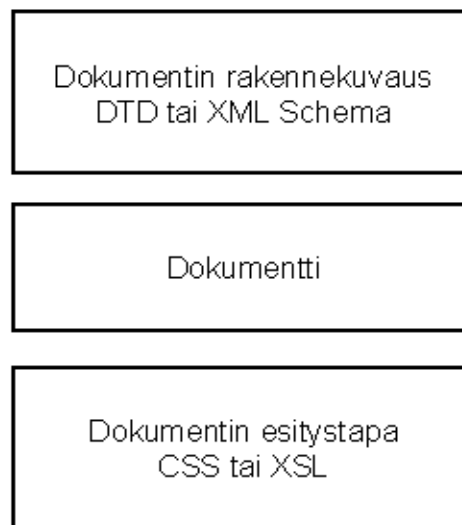
Jäsentäjä tarkistaa XML-dokumentin ja välittää sen XML-sovellukselle, jos dokumentti täyttää tarvittavat vaatimukset (Lipitsäinen, 2001). Kuviossa 14 on esitetty, kuinka jäsentäjä välittää dokumentin sovellukselle.



KUVIO 14. XML-dokumentin osat (Lipitsäinen 2001)

XML-dokumenttiin voidaan liittää myös DTD (Document Type Definition) eli rakennekuvaus (W3C, 2003). Rakennekuvaus kuvaa dokumentin rakenteen, esimerkiksi mistä elementeistä dokumentti koostuu. Elementin sisäinen rakenne osoitetaan elementin tyyppimäärittelyn avulla.

XML-dokumentti on rakenteinen. Rakenteisessa dokumentissa rakenne, sisältö ja esitystapa on erotettu toisistaan (Lipitsäinen, 2001). Kuviossa 15 on esitetty XML-dokumentin osat.



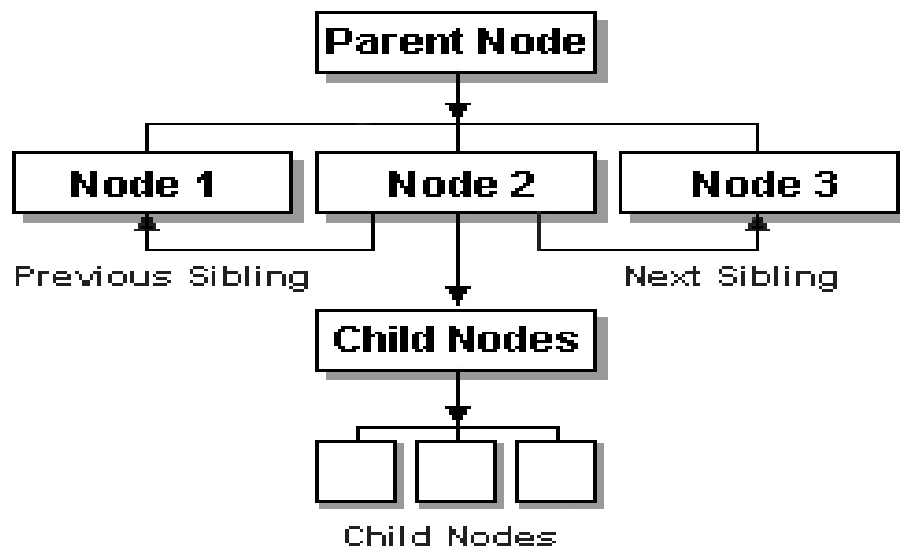
KUVIO 15. Rakenteinen dokumentti (Lipitsäinen 2001)

3.6 DOM JA SAX

DOM (Document Object Model) eli Dokumentti Oliomalli-standardi määrittelee ohjelmointikieliriippumattoman dokumenttioliomallin HTML- ja XML-dokumentteihin. DOM-mallin avulla voidaan muuttaa dokumentin sisältöä, tyyliä sekä elementtien järjestystä (Peltomäki 2001, 243-260). DOM-mallissa tieto esitetään puumaisena tietorakenteena.

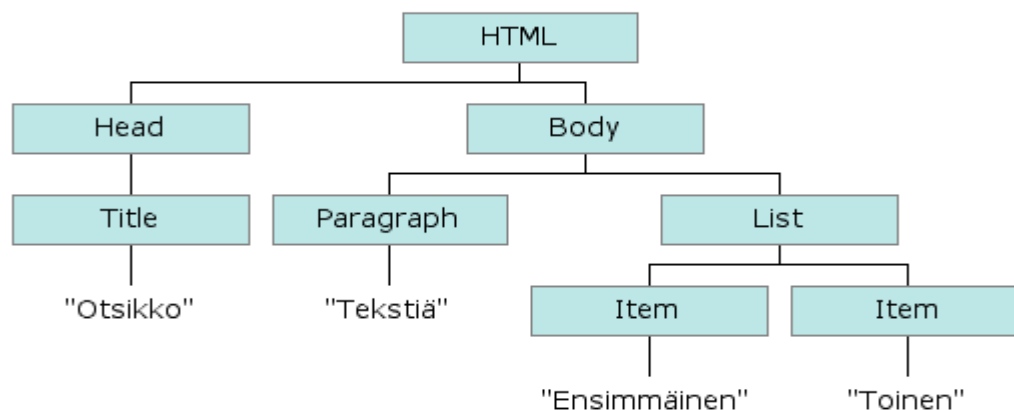
Tieto määrittelee, kuinka dokumentin elementit kommunikoivat keskenään sekä kuinka elementteihin voidaan viitata (W3C, 1997-2005). DOM-oliomalli koostuu solmuista.

Dokumenttipuu ja solmut on esitetty kuviossa 16. Jokaisella solmulla voi olla useita lapsisolmuja (child node). Kaikilla paitsi juurisolmulla on äitisolmu (parent node). Dokumenttipuussa voidaan liikkua vapaasti, ja se voidaan kulkea läpi viittaamalla mihin tahansa solmun lapseen; vanhempaan tai sisareen. Jokainen dokumentin solmu (elementti) on oma olionsa ja sisältää ominaisuudet ja metodit (2kmediat, 2000-2008). DOM-mallin mukainen jäsentäjä lukee XML-tiedoston tietokoneen muistiin puumuodossa.



KUVIO 16. Dokumenttipuu (Microsoft 2008)

DOM-malli sisältää kuvauksen dokumentin muodostavista elementeistä, elementin toiminnasta ja ominaisuuksista sekä elementtien välisistä yhteyksistä. Kuviossa 17 on esitetty dokumenttipuu HTML-tiedostosta.



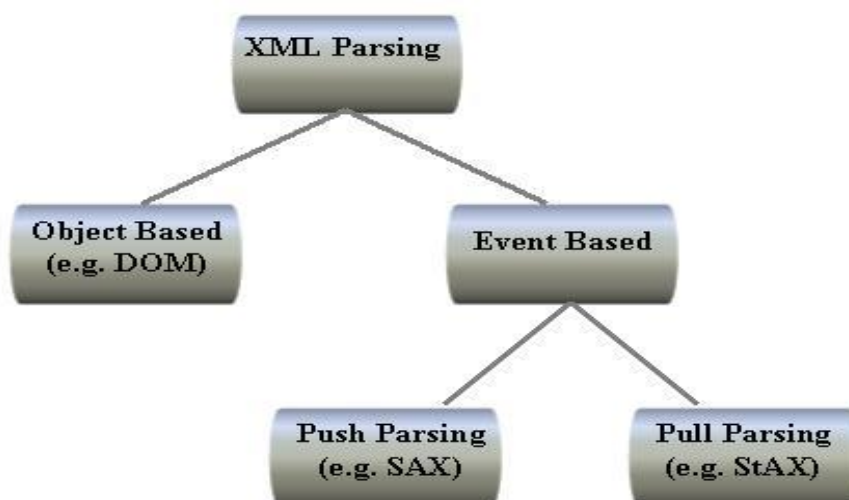
KUVIO 17. DOM-malli (Leponiemi 2002)

SAX (Simple API for XML) on yksinkertainen rajapinta XML:lle, joka perustuu tapahtumapohjaiseen käsittelyyn (Lahtonen, 2003). SAX on huomattavasti tehokkaampi kuin DOM, mutta vaikeampikäyttöinen (Helsingin Yliopisto, 2007).

SAX:in käyttö on järkevää, jos sovellus on nopeuskriittinen. SAX:in lähdekoodi on avointa, ja se on saatavilla useimpiin ohjelmointikieliin (SAX project, 2001-2008). Kuviossa 18 on esitetty myös StAX, joka pyrkii yhdistämään SAX:in tehokkuuden ja DOM:in helppokäyttöisyyden. SAX toimii ”push parsing”-periaatteella, jolloin XML parser lähettää asiakkaalle tietoa, StAX taas on ”pull parsing”-tyyppinen, jolloin asiakas hakee tietoa tarvittaessa (Tampereen teknillinen yliopisto, 2007) (Megginson technologies, 1998-2008). SAX:in ja DOM:in ongelmana on kuitenkin se että asiakas (client) ei pysty käytännössä vaikuttamaan jäsenysprosessiin (parsing).

Vaikka SAX tai StAX ei luekaan koko tiedostoa kerralla muistiin; jäsenysprosessi on yksisuuntainen, joten jo aikaisemmin luettua dataa ei voida lukea uudelleen käynnistämättä prosessia uudelleen alusta. StAX mahdollistaa prosessin käynnistämisen lisäksi jäsenysprosessin pysäyttämisen, jatkamisen ja uudelleen käynnistämisen. Tästä syystä StAX on erittäin käyttökelpoinen varsinkin muisti- ja suorituskykykriittisissä sovelluksissa (Sun Microsystems, 1995-2008). StAX:ia käytettäessä ohjelmoija pystyy vaikuttamaan jäsenysprosessiin ja ohjelmoija voi pyytää jäsentäjältä seuraavaa tapahtumaa.

StAX eroaa SAX:ista myös siinä että se on kaksisuuntainen, jäsentäjän avulla voi lukea sekä kirjoittaa XML-dokumentteja (Sun Microsystems, 2006).



KUVIO 18. XML Parsing (O'Reilly Media Inc 2005)

DOM on käyttökelpoinen varsinkin sovelluksiin, jotka järjestävät, poistavat tai lisäävät elementtejä. SAX taas soveltuu sovelluksiin, jotka eivät muuta XML-puuta, tällaisia ovat mm. hakusovellukset. DOM-jäsentäjää ei voida käyttää esimerkiksi XML-muotoa käyttävissä reaaliaikaisissa chat-sovelluksissa, koska asiakkaan ja palvelimen välinen tietovirta koostetaan käyttäjien syötteistä. DOM-jäsentäjän käyttö aiheuttaisi tällaiseen sovellukseen viiveitä, muistiin lukemisen vuoksi.

SAX- ja StAX-jäsentäjien käyttö mahdollistaa periaatteessa rajattoman tiedostopituuden lukemisen, koska tapahtumat haetaan tiedostosta yksi kerrallaan tietokoneen muistin määrä ei vaikuta jäsenysprosessiin (IBM, 2006).

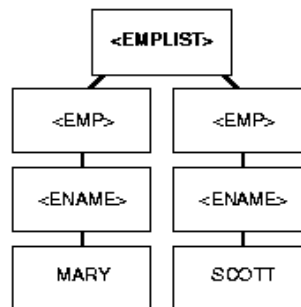
Kuviossa 19 on näytetty kuinka DOM luo XML-dokumentista puun ja SAX luo elementeistä lineaarisen listan. DOM-mallin luoma puu koostuu juurisolmusta document, äitisolmusta <EMPLIST>, sisarus- ja lapsisolmuista <EMP>, sisarus- ja lapsisolmuista <ENAME> sekä sisarus- ja lapsisolmuista MARY ja SCOTT.

SAX:in luoma lineaarinen lista taas koostuu tapahtumajonosta, tapahtumia ovat: dokumentin alku, elementin alku EMPLIST, elementin alku EMP, elementin alku ENAME, merkkidata MARY, elementin loppu EMP, elementin alku EMP, elementin alku ENAME, merkkidata SCOTT, elementin loppu EMP, elementin loppu EMPLIST ja dokumentin loppu.

XML Document

```
<?XML Version = "1.0"?>
<EMPLIST>
  <EMP>
    <ENAME>MARY</ENAME>
  </EMP>
  <EMP>
    <ENAME>SCOTT</ENAME>
  </EMP>
</EMPLIST>
```

The DOM interface creates a TREE structure based on the XML Document



Useful for applications that include changes eg. reordering, adding, or deleting elements.

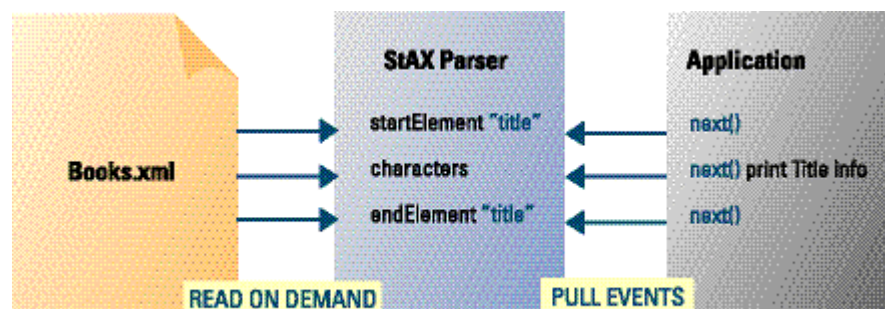
The SAX interface creates a series of linear events based on the XML document

```
start document
start element: EMPLIST
start element: EMP
start element: ENAME
characters: MARY
end element: EMP
start element: EMP
start element: ENAME
characters: SCOTT
end element: EMP
end element: EMPLIST
end document
```

Useful for applications such as search and retrieval that do not change the 'XML tree'.

KUVIO 19. DOM ja SAX (Oracle)

Kuviossa 20 on esitetty StAX-jäsentäjän toimintaa. Sovellus pyytää StAX-jäsentäjää lähettämään seuraavan tapahtuman, StAX-jäsentäjä hakee tapahtuman Books.xml tiedostosta ja välittää sen sovellukselle.

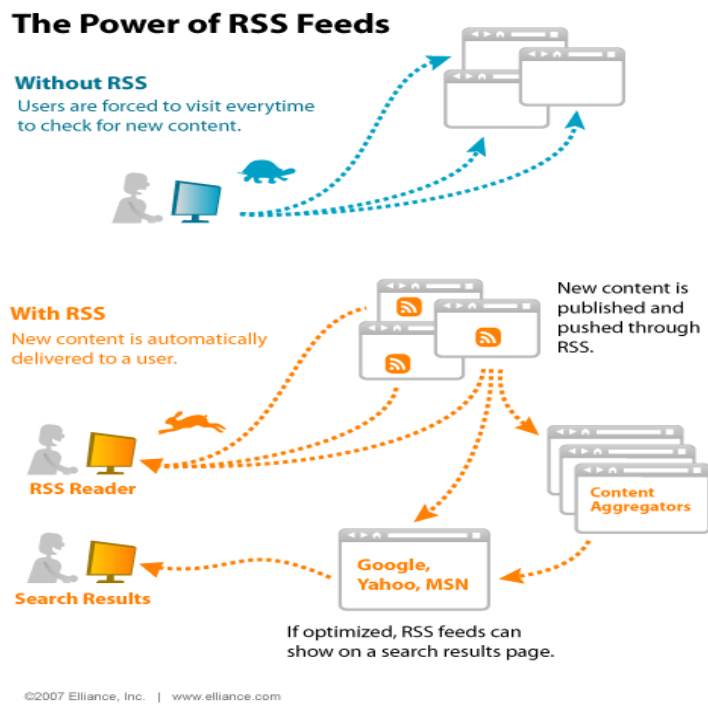


KUVIO 20. Ohjelma pyytää StAX:ilta tapahtumaa (Oracle 2003)

3.7 RSS

RSS (Really Simple Syndication tai Rich Site Summary) eli uutissyöte on XML-pohjainen määrittely. Tiedon lähettäjä muuntaa tiedon RSS-muotoon ja vastaanottaja saa tiedon uudesta tekstistä lukijaohjelmalla. RSS-tekniikan avulla voi siten tarkkailla www-sivulla tapahtuvia muutoksia, ilman että sivustoa tarvitsee jatkuvasti tarkkailla.

RSS-lukuohjelmia on useita: maksullisia sekä maksuttomia. Myös osa www-selaimista tukee RSS:ää. RSS lähettää käyttäjälle vain käyttäjän tilaamat syötteet (Talentum, 1998-2007). Kuviossa 21 esitetään www-sivun muutosten seuraamista RSS:n avulla ja ilman.



KUVIO 21. RSS-syöte (Elliance Inc. 2007)

4 TOTEUTUS

4.1 Monket-Calendar

Monket-Calendar on Karl O'Keeffe:n tekemä kalenterisovellus. Monket-Calendar on tehty PHPiCalendarin pohjalta. Monket-Calendar perustuu GPL (General Public License) ja CC (Creative Commons license) lisensseihin. Tässä työssä tutkittiin Monket-Calendarin 0.9.1 versiota, joka on vielä betavaiheessa eikä toiminut sellaisenaan. Työn tekohetkellä Monket-Calendarin kehitys oli pysähtynyt, eivätkä myöskään projektin www-sivut eivät olleet toiminnassa. Monket-Calendar on toteutettu pääosin AJAX:ia käyttäen.

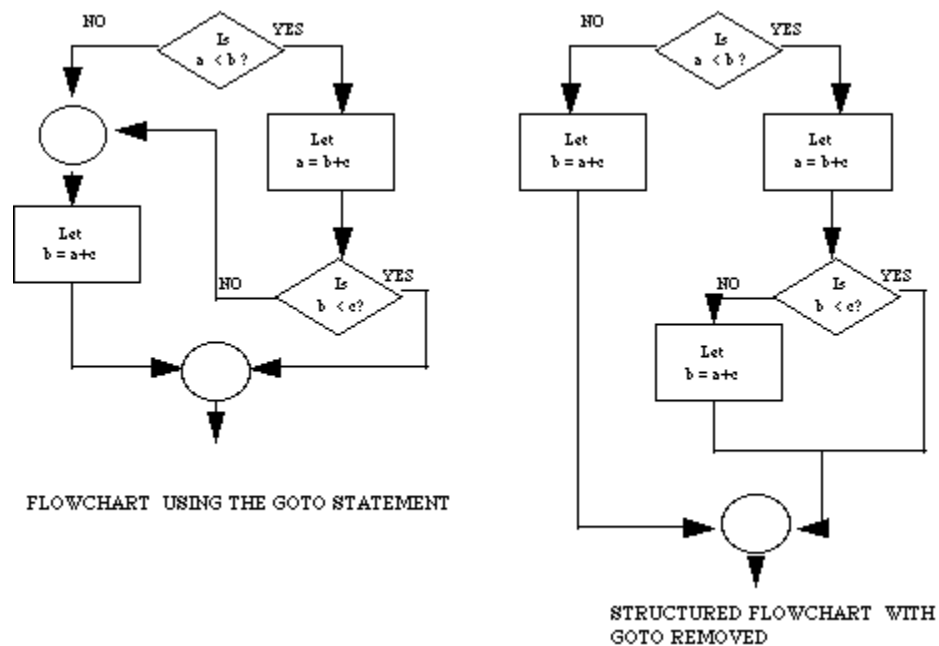
Monketin parasta antia on kuitenkin sen käyttöliittymä, joka on pyritty tekemään mahdollisimman helppokäyttöiseksi. Monketin käyttöliittymä on esitetty kuviossa 22.



KUVIO 22. Monket Calendar

Monket käyttää seuraavia kirjastoja pohjana: PHPiCalendar .ics tiedostojen jäsenyykseen, DataRequestor Ajax kutsuihin, ie7 Internet Explorer 7 tukeen, Dom-Drag javascript vedä ja tiputa tukeen ja Dom-Tooltip javascript työkalu tukeen (Keefe, 2003).

Monket-Calendar on toteutettu rakenteista ohjelmointia käyttäen. Rakenteisia ohjelmia on helppo ymmärtää, helppo muuttaa ja helppo testata (Silander, 1999). Kuten kuvion 23 oikean reunan kuviosta voidaan nähdä, rakenteisella logiikalla ohjelmoidussa ohjelmassa on selkeä alku- ja loppupiste. Ei-rakenteista ohjelmointikoodia kutsutaankin usein nimellä ”spagettikoodi”, jolloin ohjelmalla ei ole selkeää alku- ja loppupistettä ja ohjelmassa saattaa olla useitakin haaroja keskellä ohjelmakoodia. Näistä haaroista johtuen tällainen koodi on vaikeata tutkia ja seurata.



KUVIO 23. Rakenteinen ja ei-rakenteinen ohjelmointi (Avanija 2007)

4.2 Työn toteutus

Koska tarkoituksena on hallita monen työntekijän työlistaa yhdessä kalenterinäkyvässä, täytyy muokattavaa kalenteriluokkaa pystyä muuttamaan helposti. Monket ei antanut mahdollisuutta muuttaa muokattavaa kalenteria, vaan muokattava kalenteri oli määritelty kiinteästi asetustiedostossa.

Kalenterinäkyvään lisättiin lista, josta voidaan dynaamisesti vaihtaa muokattavaa kalenteriluokkaa sekä tulostettiin ruudulle tällähetkellä muokattavan kalenteriluokan nimi.

Koska työt saattavat olla useamman päivän mittaisia, on tärkeää, että varaus voidaan venyttää ruudulla useammalle päivälle. Monket tukee tätä ominaisuutta vakiona. Eri kalenteriluokille tai henkilöille varatut työt näkyvät eri väreillä, mikä lisää käyttöliittymän selkeyttä. Eri päiville varattuja töitä pystyy myös helposti siirtämään toiselle päivälle, hiirellä vetämällä.

Uusia kalentereita voidaan luoda luomalla .ics tiedosto kalenteri kansioon esimerkiksi Microsoft Outlookin kalenterilla. Tämän jälkeen Monket ottaa kalenterin automaattisesti käyttöön, jos kalenteritiedosto sijaitsee oikeassa kansiossa.

Kalentereista ei poisteta historiatietoja, vaan historiatiedot ovat saatavilla koko kalenterien olemassaolon ajalta, jos käyttäjä ei niitä erikseen poista. Eri kalenteriluokat sijaitsevat omissa .ics-tiedostoissaan kalenterikansiossa.

Tämä ei kuitenkaan muodostu nykyisten levytilojen kanssa ongelmaksi, koska .ics tiedostot ovat ASCII (American Standard Code For Information Interchange) muodossa ja ovat näin kooltaan erittäin pieniä. Jos kalenteritiedostot ovat kovin pitkiä, ongelmaksi saattaa kuitenkin muodostua suorituskyky.

Jos kuitenkin joskus haluaa varmuuskopioida vanhat tapahtumat ja luoda uuden kalenterin, käy tämä kätevimmin kopioimalla vanha .ics tiedosto talteen ja luomalla uusi samanniminen .ics tiedosto vanhan tilalle.

Kalenterin päivittäinen varmuuskopiointi on toteutettu automaattisella ajastuksella erilliselle verkkolevylle. Kalenterimerkintöjen palauttaminen varmistustiedostosta tapahtuu kopioimalla varmistustiedosto vioittuneen tai kadonneen tiedoston paikalle.

Ulkoasu on toteutettu CSS tyyლისivujen avulla ja ympäristömuuttujat määritellään erillisessä konfiguraatio tiedostoissa. Tiedostot on jaettu selvyuden vuoksi eri kansioihin tiedostojärjestelmässä tiedostotyyppien mukaan. Kalenteria voi selata kuukausi kerrallaan navigointi nuolien avulla.

Uusia merkintöjä luodaan valitsemalla kalenteriluokka, sitten painetaan ”+” merkkiä päivämäärässä, tämän jälkeen kirjoitetaan merkintä kalenteriin. Koska kalenteri on luotu dynaamiseksi web-sovellukseksi, uudet kalenterimerkinnät näytetään kalenterissa heti, ilman uudelleenlataamista.

Nykyinen aika haetaan PHP:n ”getdate” funktiolla, funktio palauttaa ajan kuviossa 24 esitettyssä muodossa assosiatiiviseen taulukkoon

```
[seconds] => 40
[minutes] => 58
[hours]   => 21
[mday]    => 17
[wday]    => 2
[mon]     => 6
[year]    => 2003
[yday]    => 167
[weekday] => Tuesday
[month]   => June
[0]       => 1055901520
```

KUVIO 24. Getdate-funktio (The PHP Group 2001-2008)

Alin luku [0] on ns. ”unix time”, joka kertoo, montako sekuntia on kulunut keskiyöstä 1.1.1970 tammikuuta alkaen.

Kalenteriin luotiin viikkonäkymä kuluvalta viikolta. Viikkoja voi selata ruudulla näkyvien nuolien avulla tai syöttämällä päivämäärän. Kuviossa 25 on esitetty kalenterin käyttöliittymä.



KUVIO 25. Kalenterin käyttöliittymä

Kalenterimerkintöihin tulostetaan automaattisesti myös kalenterin nimi ja rivinvaihto merkinnän lisäksi. Kalenterimerkinnän teksti rivitetään automaattisesti, jos merkintä ei mahdu yhdelle riville.

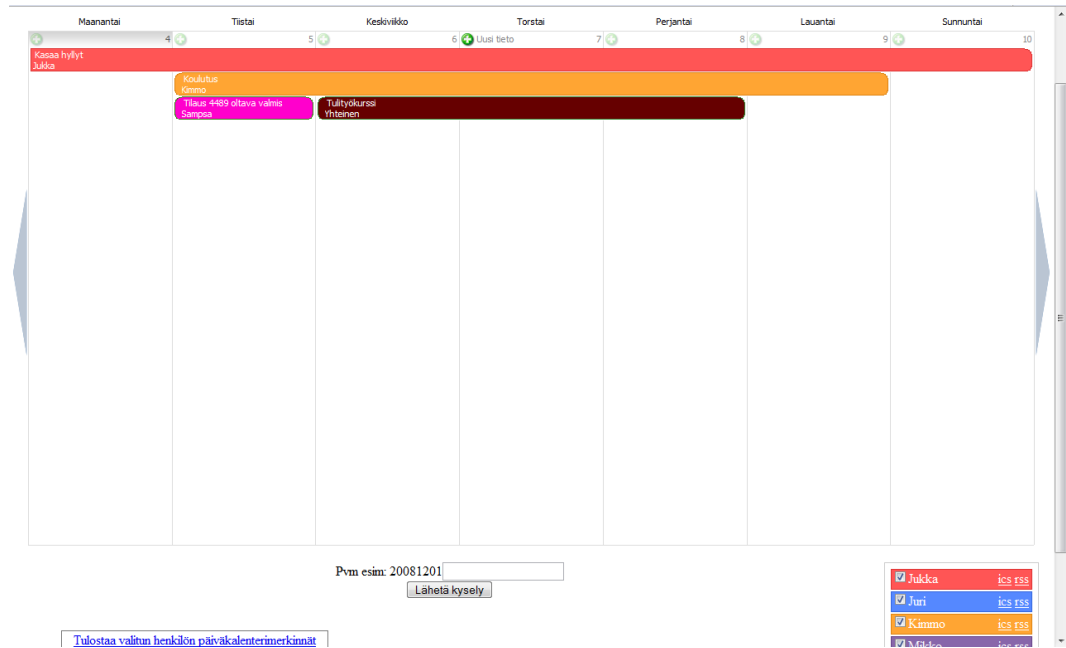
Kalenterimerkintöjen värit on määritetty css tyylitiedostossa. Kalenteriluokat ja niiden värit näytetään kuviossa 26 esitetystä listasta. Lista on sijoitettu ruudulla kalenterien alapuolelle oikeaan alanurkkaan.



KUVIO 26. Kalenterilista

Kalenterissa on myös valmius käyttäjätunnistukselle ja -rajaukselle tulevaisuutta varten. Näin voidaan määritellä, mitä kalentereja kukin käyttäjä voi käsitellä. Kalenteritapahtumat voidaan välittää myös RSS-muodossa, valintalistalla olevilla painikkeilla.

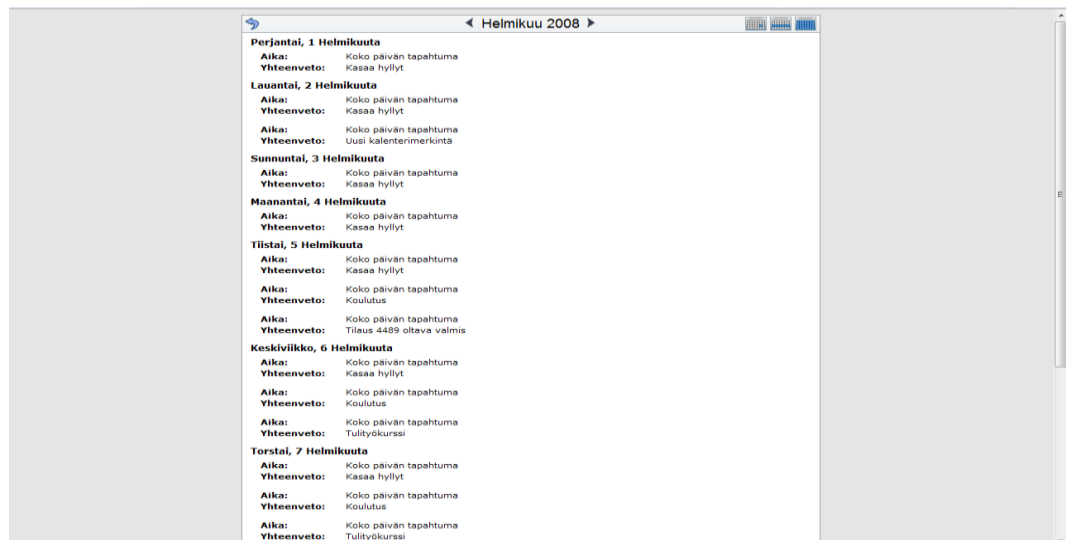
Kalenterimerkintä luodaan painamalla ”+” –painiketta halutun päivämäärän kohdalla tai kaksoisnapauttamalla tyhjällä alueella. Sovellus luo valitun päivän kohdalle muokattavan kalenterin värisen laatikon, johon kirjoitetaan merkinnän teksti. Tämän jälkeen merkintää voi siirtää tai venyttää useammalle päivälle vapaasti. Kuviossa 27 näytetään kuinka kalenterimerkinnät venytetään useamman päivän pituisiksi. Kalenterimerkintöjä voidaan venyttää niin pitkäksi kuin on tarvetta, tietysti ruudulla näkyvien päivien rajoittamissa rajoissa.



KUVIO 27. Uuden tiedon luominen ja merkintöjen venytys usealle päivälle

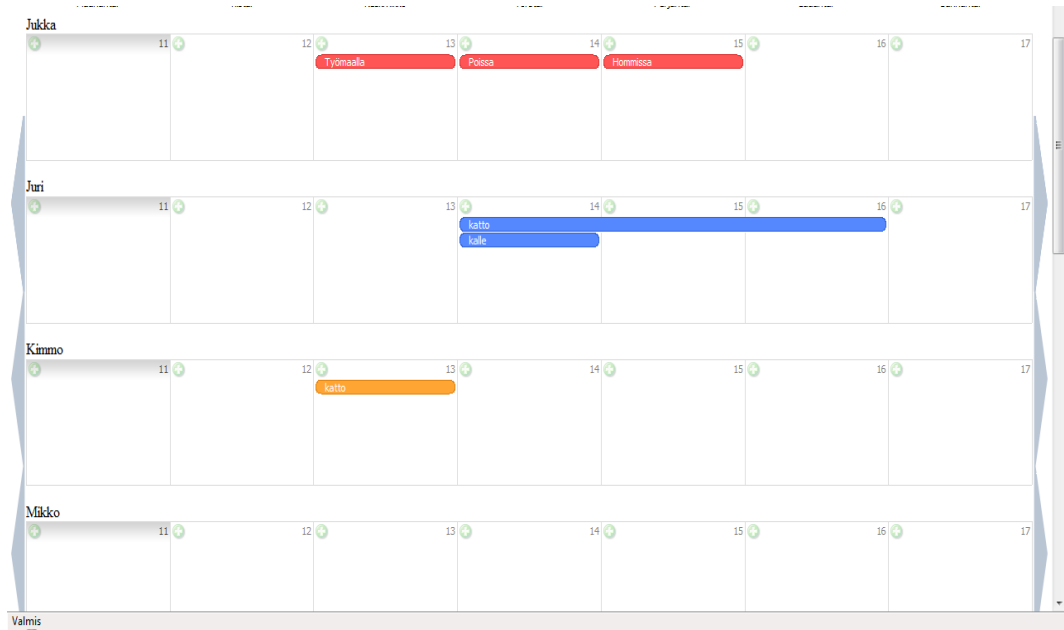
Kalenterisivun alareunassa on linkkilista, josta voi tulostaa ruudulle valitun kalenteriluokan tai kaikkien kalenterien tiedot päivältä, viikolta tai kuukaudelta.

Kuvion 28 esittämästä muodosta kalenterimerkinnot on hyvä tulostaa esimerkiksi paperiversioksi arkistoon.



KUVIO 28. Tapahtumien tulostusmuoto

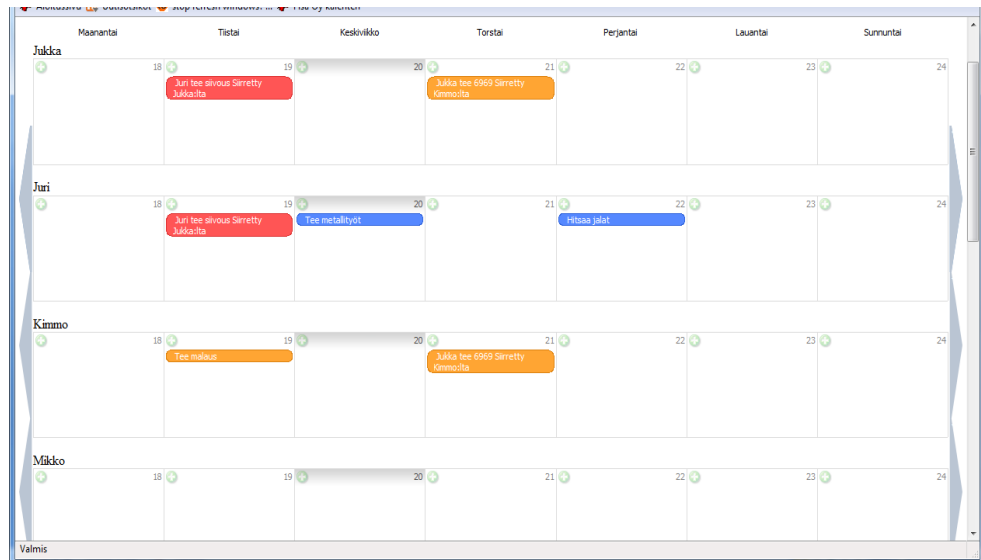
Näkymä päätettiin kuitenkin muuttaa erilaiseksi. On helpompi tulkita tyhjiä aikoja kalenterissa, kun jokaiselle kalenteriluokalle on oma näkymänsä (kuvio 29).



KUVIO 29. Kalenterinäkymä

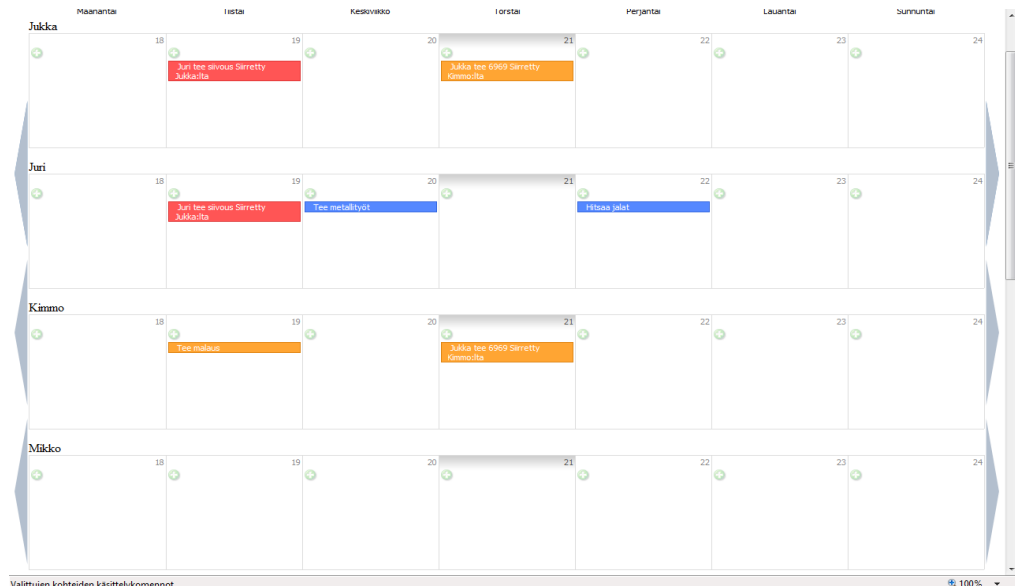
Osa kalenterin toiminnoista on toistaiseksi toteutettu AJAX:illa ja osa taas perus PHP:llä, jolloin selain lataa sivun uudelleen käyttäjän tapahtuman jälkeen. Monket-Calendar on aluperinkin jaettu useisiin tiedostoihin, esimerkiksi JavaScript funktiot ovat omissa tiedostoissaan. Tämä rakenne säilytettiin, koska se helpottaa koodin tutkimista ja jatkokehitystä. Tiedostot on jaettu myös kansioihin tiedostotyyppin mukaan, esimerkiksi css- ja konfiguraatitiedostoille on omat kansionsa.

Koska joskus voi olla tarvetta siirtää töitä henkilöltä toiselle, lisättiin kalenteriin myös toiminto, joka kopioi työn toiselle henkilölle. Kopio luodaan kalenterimerkinnässä määriteltyyn kalenteriluokkaan automaattisesti, kun kalenterimerkinnän teksti sisältää jonkin toisen kalenteriluokan nimen (kuvio 30). Kopio luodaan kaikkiin kalenteriluokkiin siinä tapauksissa, että kalenterimerkinnän teksti sisältää sana yhteinen (ks. LIITE 1).



KUVIO 30. Merkintöjen siirto eri kalenteriluokkiin

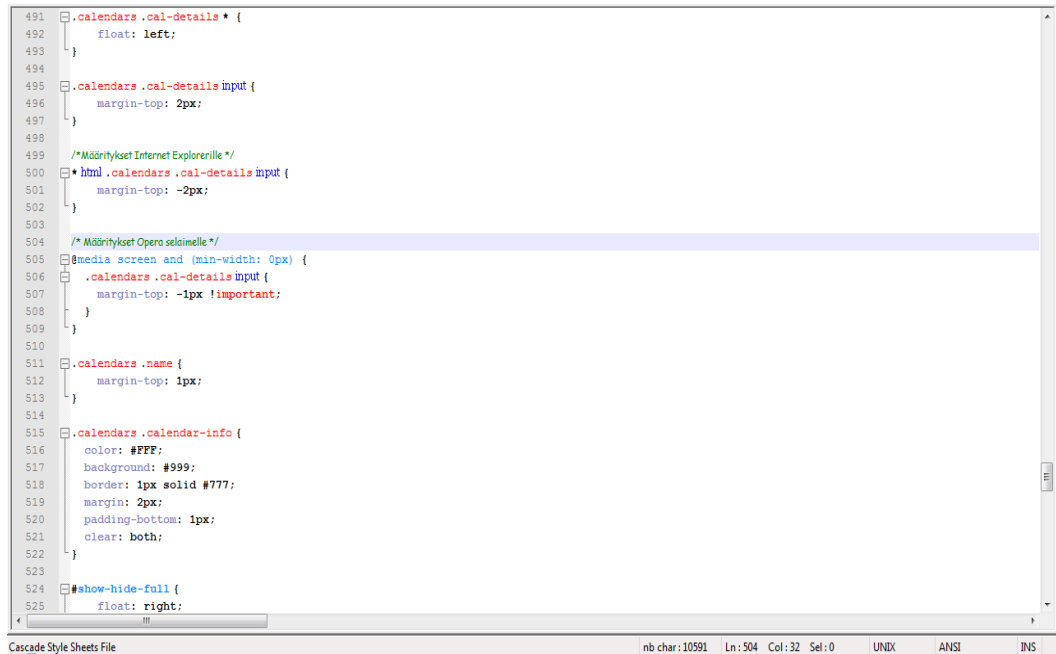
Kalenteri on hieman eri näköinen eri selaimilla johtuen eroista selaimien CSS kielen tulkinnessa. Yllä oleva kuvio 30. on Mozilla Firefox 2 selaimesta ja alla oleva kuvio 31. Internet Explorer 7 selaimesta.



KUVIO 31. Kalenteri IE7 selaimessa

Kuten kuviosta 31. voidaan huomata, kalenterimerkintöjen reunat eivät ole pyöristettyjä Internet Explorer selaimissa.

Myös pienempiä eroja ulkoasussa löytyy, mm. css. määrittelyihin on tehty eri selaimille erilaisia merkintöjä (kuvio 32). Muutokset koskevat lähinnä Opera ja Internet Explorer selaimien ongelmia css-määriteltujen elementtien marginaaleissa, ko. selaimet eivät täysin noudata CSS-standardeja.



```

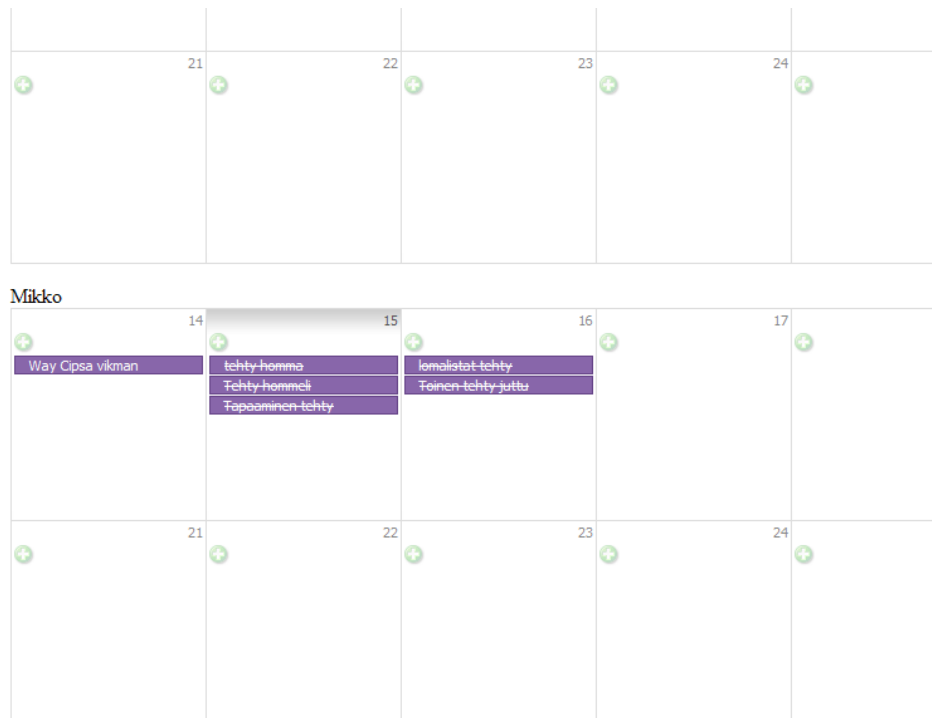
491 .calendars .cal-details * {
492     float: left;
493 }
494
495 .calendars .cal-details input {
496     margin-top: 2px;
497 }
498
499 /*Määrittely Internet Explorerille*/
500 *html .calendars .cal-details input {
501     margin-top: -2px;
502 }
503
504 /* Määrittely Opera selaimelle */
505 @media screen and (min-vidch: 0px) {
506     .calendars .cal-details input {
507         margin-top: -1px !important;
508     }
509 }
510
511 .calendars .name {
512     margin-top: 1px;
513 }
514
515 .calendars .calendar-info {
516     color: #FFF;
517     background: #999;
518     border: 1px solid #777;
519     margin: 2px;
520     padding-bottom: 1px;
521     clear: both;
522 }
523
524 #show-hide-full {
525     float: right;

```

Cascade Style Sheets File nb char: 10591 Ln: 504 Col: 32 Sel: 0 UNIX ANSI INS

KUVIO 32. Calendar.css tiedosto

Lopuksi muutettiin vielä näkymä kaksiviikkoiseksi sekä tehtiin kalenteriin toiminto, joka yliviivaa tekstin, jos kalenterimerkinnässä on sana ”tehty”. Yliviivattu kalenterimerkintä on helppo tunnistaa tehdyksi työksi, näin työjonon seuraaminen helpottuu. Tehtyjä kalenterimerkintöjä on esitetty kuviossa 33.



KUVIO 33. Kalenterimerkintöjen merkitseminen tehdyksi

5 YHTEENVETO

Opinnäytetyön tavoitteena oli tutkia ja toteuttaa kalenterisovellus pienyrityksen käyttöön. Kalenteri saatiin toteutettua ulkoasultaan halutun näköiseksi, ja halutut toiminnot saatiin toteutetuksi kalenteriin. Tutkimustyön avulla saatiin myös paljon tietoa, jota käytetään tulevaisuudessa hyödyksi, kun sovellusta jatkokehitetään edelleen. Tutkimuksessa havaittiin AJAX-tekniikan sopivan varsin hyvinä tämänkaltaisiin sovelluksiin. AJAXin avulla sovellukset saadaan vaikuttamaan työpöytäsovelluksilta. Käyttäjän toimet vaikuttavat näytölle heti, eikä sivua tarvitse ladata uudelleen. Näin sovellus on helppo ja nopea käyttää. Tulevissa versioissa onkin tarkoitus siirtää loputkin, vielä PHP:llä toteutetut toiminnot AJAX-pohjaisiksi.

Opinnäytetyö täytti sille asetetut tavoitteet kalenterisovelluksen toteuttamisesta. Kalenterisovellus on otettu käyttöön Hollolan Sähköautomaattikka Oy:ssä. Käyttäjien kommentit ovat olleet pääasiassa positiivisia, mutta joitakin muutoksia on kuitenkin toivottu. Nämä muutokset tullaan huomioimaan kalenterisovelluksen seuraavassa kehitysversiossa. Toivottuja muutoksia ovat olleet uuden kalenterimerkinnän asettelu merkintää luotaessa sekä viikkojen selauspainikkeiden sijoittelu.

Monket-Calendarin kehittäjän kehitystyö näyttää pysähtyneen, mutta jotkin kaupalliset palveluntarjoajat ovat ottaneet Monket-Calendarin alustakseen. PHPiCalendarin kehitys sen sijaan näyttäisi jatkuvan aktiivisena.

LÄHTEET

2kmediat. 2000-2008. PHP [verkkajulkaisu]. KK Mediat [viitattu 14.2.2008].

Saatavissa: <http://www.2kmediat.com/php/johdanto.asp>

2kmediat 2000-2008. CSS:n laatikkomalli ja asemointi [verkkajulkaisu]. [viitattu

8.2.2008]. Saatavissa: <http://www.2kmediat.com/css/lohkot.asp>

Avanija 2007. VC++ tutorial for beginners [verkkajulkaisu]. Connexions [viitattu

14.03.2008]. Saatavissa: <http://cnx.org/content/m14425/latest/>

Darie, C. 2006. AJAX with PHP tutorial [verkkajulkaisu]. Packt Publishing

[viitattu 13.2.2008]. Saatavissa: <http://www.cristiandarie.ro/ajax-php/>

Dawson, F & Stenerson, D. 1998. Request For Comments rfc 2445 [verkkajulkai-

su]. IETF [viitattu 1.2.2008]. Saatavissa: <http://www.ietf.org/rfc/rfc2445.txt>

Eclipse Foundation 2008. Eclipse IDE rakenne [verkkajulkaisu]. Eclipse project

[viitattu 12.2.2008]. Saatavissa: <http://www.eclipse.org/proposals/php-ide/>

Ek, J. & Eriksson, U. 2001. Linux Internet palvelimena käsikirja. 1. Painos.

Vantaa: Schildts kustannus Oy

Elliance Inc. 2007. Power of RSS [verkkajulkaisu]. [viitattu 14.2.2008].

Saatavissa: <http://www.elliance.com>

Google 2008. What is an iCal file? [verkkajulkaisu]. [viitattu 2.2.2008].

Saatavissa: <http://www.google.com/support/calendar/bin/answer.py?hl=en-ch&answer=45664>

Helsingin Yliopisto 2007. XML-ohjelmointirajapinnat [verkkajulkaisu]. Helsinki

University of Technology, Media Technology [viitattu 14.2.2008]. Saatavissa:

http://www.media.hut.fi/as011110/kalvot_2007/6-ohjelmointi.pdf

IBM 2006. StAX'ing up XML [verkkajulkaisu]. [viitattu 12.4.2008]. Saatavissa: <http://www.ibm.com/developerworks/library/x-stax1.html>

Karl'o Keeffe. 2003. Monket-Calendar Wiki [verkkajulkaisu]. [viitattu 13.2.2008]. Saatavissa: http://monket.net/wiki-v2/Monket_Calendar

Kollanus, S. 2005. WWW-sovellukset [verkkajulkaisu]. Jyväskylän yliopisto [viitattu 10.2.2008]. Saatavissa: http://www.cs.jyu.fi/~kolli/ITK215_05/

Korpela, J. 2007. Javascript web-julkaisemisen opas [verkkajulkaisu]. [viitattu 4.2.2008]. Saatavissa: <http://www.cs.tut.fi/~jkorpela/webjulk/3.2.html>

Korpela, J. 2007. Rakennetta mukaan (web julkaisemisen opas) [verkkajulkaisu]. [viitattu 2.2.2008]. Saatavissa: <http://www.cs.tut.fi/~jkorpela/webjulk/1.2.html>

Lahtonen, T., Ekonoja, A., Mäntylä, J. & Heinonen, P. 2007. WWW-sovellukset [verkkajulkaisu]. Jyväskylän avoin yliopisto [viitattu 26.12.2007]. Saatavissa: <http://appro.mit.jyu.fi/sovellukset/>

Lahtonen, T. 2003. XML-verkkotekniikan jatkokurssi [verkkajulkaisu]. Jyväskylän avoin yliopisto [viitattu 5.2.2008]. Saatavissa: <http://appro.mit.jyu.fi/2003/kevat/vtj/luennot/luento10/>

Lehto, T. 2007. Dynaaminen web-palvelin helposti käyttöön Linuxissa [verkkajulkaisu]. Sanoma Magazines Finland [viitattu 2.2.2008]. Saatavissa: <http://www.tietokone.fi/softa/tiedosto.asp?id=13140>

Lehtonen, K. 2001. HTML-opas [verkkajulkaisu]. Helsingin ammattikorkeakoulu [viitattu 2.2.2008]. Saatavissa: <http://cs.stadia.fi/~lehtk/html.html>

Leponiemi, J. 2002. Javascript ja DHTML [verkkajulkaisu]. Tampereen yliopisto [viitattu 1.2.2008]. Saatavissa: <http://www.uta.fi/~jl/javascript/>

Lipitsäinen A. 2001. XML – Extensible Markup Language [verkkajulkaisu]. Helsingin liiketalouden ammattikorkeakoulu [viitattu 14.2.2008]. Saatavissa: <http://myy.helia.fi/~atk42d/Xml/xmlopas.html>

Megginson technologies Ltd. 1998-2008. Simple API for XML [verkkajulkaisu]. [viitattu 10.2.2008]. Saatavissa: <http://www.megginson.com/downloads/SAX/>

Microsoft 2008. MSDN Library [verkkajulkaisu]. Microsoft [viitattu 22.1.2008]. Saatavissa: <http://msdn2.microsoft.com/en-us/library/default.aspx>

Netscape 1996. Javascript handbook [verkkajulkaisu]. Netscape [viitattu 13.1.2008]. Saatavissa: <http://wp.netscape.com/eng/mozilla/3.0/handbook/javascript/>

Oracle 2003. Parsing XML efficiently [verkkajulkaisu]. [viitattu 12.4.2008]. Saatavissa: <http://www.oracle.com/technology/oramag/oracle/03sep/o53devxml.html>

O'Reilly Media Inc. 2008. XML-Parsing [verkkajulkaisu]. [viitattu 16.2.2008]. Saatavissa: <http://www.xml.com/2005/07/06/graphics/image001.jpg>

Oulun seudun ammattikorkeakoulu. 2003-2006. Web-sovellusten ohjelmointi [verkkajulkaisu]. Software Business Competence [viitattu 4.1.2008]. Saatavissa: <http://www.oamk.fi/sbc/www/johdanto.php>

Oulun seudun ammattiopisto. 2006. HTML- ja PHP-ohjelmointi [verkkajulkaisu]. [viitattu 17.2.2008]. Saatavissa: http://materiaali.osao.fi/kaul/verkko-opetus/datanomi/ammattilliset_valinnaiset/html_ja_php_ohjelmointi/perusteet.php

Peltomäki, J. 2001. Javascript Toolkit. 1. Painos. Jyväskylä: Docendo Finland Oy.

Peltomäki, J. 1998. WWW-ohjelmointi. 1.Painos. Jyväskylä: Teknolit Oy.

Plaze 2007. Plazes everywhere [verkkojulkaisu]. [viitattu 11.03.2008]. Saatavissa: <http://blog.plazes.com>

Rantala, A. 2002. PHP web ohjelmoinnin peruskirja. 1. Painos. Porvoo: WS BookWell.

SAX project 2001-2008. SAX [verkkojulkaisu]. [viitattu 9.2.2008]. Saatavissa: <http://www.saxproject.org/>

Seidler, K. 2002-2008. Xampp [verkkojulkaisu]. Apache friends [viitattu 3.2.2008]. Saatavissa: <http://www.apachefriends.org/en/xampp.html>

Silander 1999. Rakenteinen ohjelmointi [verkkojulkaisu]. Helsingin ammattikorkeakoulu Stadia [viitattu 14.03.2008]. Saatavissa: http://cs.stadia.fi/~silander/ohjelmointi/c_opas-Rakentei.html

Smith, D. & Negrino, T. 2007. JavaScript Tehokas hallinta. 1. Painos. Jyväskylä: Gummerus Kirjapaino Oy.

Sun Microsystems 1995-2008. Introduction to StAX [verkkojulkaisu]. [viitattu 8.4.2008]. Saatavissa: <http://today.java.net/pub/a/today/2006/07/20/introduction-to-stax>

Sun Microsystems 2006. The Java Web Services Tutorial [verkkojulkaisu]. [viitattu 11.4.2008]. Saatavissa: <https://cis.med.ucalgary.ca/http/java.sun.com/webservices/docs/2.0/tutorial/doc/StAX4.html>

Talentum 1997-2008. RSS-ohje [verkkojulkaisu]. [viitattu 8.2.2008]. Saatavissa: <http://mikropc.net/ohjeet/rss.html>

Tampereen teknillinen yliopisto 2007. XML processing techniques [verkkojulkaisu]. [viitattu 13.2.2008]. Saatavissa: http://www.cs.tut.fi/kurssit/OHJ-5201/materiaali/4_XMLprocessing.pdf

The PHP Group 2001-2008. PHP: getdate [verkkajulkaisu]. [viitattu 11.03.2008].
Saatavissa: <http://fi.php.net/getdate>

W3C. 1997. Cascading Style Sheets, level 2 [verkkajulkaisu]. [viitattu 5.1.2008].
Saatavissa: <http://www.w3.org/TR/1998/PR-CSS2-19980324/>

W3C 1998. Box model [verkkajulkaisu]. [viitattu 8.2.2008]. Saatavissa:
<http://www.w3.org/TR/1998/PR-CSS2-19980324/box.html>

W3C Suomen toimisto 1999-2003. XML 10 kohdan tiivistelmä [verkkajulkaisu].
[viitattu 4.2.2008]. Saatavissa: <http://www.w3c.tut.fi/translations/xml/xmlin10opts/>

W3C 2003. XML-peruskurssi [verkkajulkaisu]. [viitattu 5.2.2008]. Saatavissa:
<http://www.w3c.tut.fi/training/2003/0219-xml-1/yhteenvedo.html>

Wenning, R. 2002. A Privacy module for web technology [verkkajulkaisu]. W3C
[viitattu 18.1.2008]. Saatavissa: <http://www.w3.org/2002/Talks/www2002-p3p/all.htm>

```

<?php

echo str_pad(' ', 4095);
flush();
ob_flush();

$current_view = "week";

define('BASE', dirname(__FILE__) . '/phpicalendar-karl/');
include(BASE.'functions/ical_parser.php');
if ($minical_view == 'current') $minical_view = 'month';

ereg ("([0-9]{4})([0-9]{2})([0-9]{2})", $getdate, $day_array2);
$this_day           = $day_array2[3];
$this_month        = $day_array2[2];
$this_year         = $day_array2[1];
$next_week         = date("Ymd", strtotime("+1 week", $unix_time));
$unix_time         = strtotime($getdate);
$today_today      = date("Ymd", strtotime("now + $second_offset
seconds"));
$tomorrows_date   = date("Ymd", strtotime("+1 day", $unix_time));
$yesterdays_date = date("Ymd", strtotime("-1 day", $unix_time));
// find out next month
$next_month_month  = ($this_month+1 == '13') ? '1' : ($this_month+1);
$next_month_day    = $this_day;
$next_month_year   = ($next_month_month == '1') ? ($this_year+1) :
$this_year;
while (!checkdate($next_month_month,$next_month_day,$next_month_year))
$next_month_day--;
$next_month_time   = mktime(0,0,0,$next_month_month,
$next_month_day,$next_month_year);
$prev_week         = date("Ymd", strtotime("-1 week", $unix_time));
// find out last month
$prev_month_month  = ($this_month-1 == '0') ? '12' : ($this_month-1);
$prev_month_day    = $this_day;
$prev_month_year   = ($prev_month_month == '12') ? ($this_year-1) :
$this_year;
while (!checkdate($prev_month_month,$prev_month_day,$prev_month_year))
$prev_month_day--;
$prev_month_time   = mktime(0,0,0,$prev_month_month,
$prev_month_day,$prev_month_year);
$next_month        = date("Ymd", $next_month_time);
$prev_month        = date("Ymd", $prev_month_time);
$display_date      = localizeDate($dateFormat_month, $unix_time);
$parse_month       = date("Ym", $unix_time);
if ($_POST["name"] != "") {define('DEFAULT_CALENDAR',
$_POST["name"]); $tmpmuuttuja = $_POST[name];} else
{define('DEFAULT_CALENDAR', "Yhteinen"); $tmpmuuttuja="Yhteinen";}
$testi             = $_REQUEST['apuri'];
if ($testi != NULL){$first_of_month = $testi;}

```

```

if ($testi == null) { $first_of_month = $this_year.$this_month.$this_day; }
//$first_of_month      = $this_year.$this_month.$testi;
$syotettyaiva        = $_POST["paivamaara"];
//$apuri = null;
if ($syotettyaiva != "")
{
$first_of_month = $syotettyaiva;
$syotettyaiva    == null;
}
//$first_of_month      = $next_week -1; //oikeasti siis viikon aloituspäivä,
joka näkyy ruudulla ekana
$start_month_day      = dateOfWeek($first_of_month, $week_start_day);
$thisday2             = localizeDate($dateFormat_week_list, $unix_time);
$num_of_events2       = 0;
function outputSingleDayEvents($daysEvents, $dayOfWeek, $daylink, $kal) {
    echo '<div class="single-day" id="day-' . $daylink . "'>';

        usort($daysEvents, cmpEventLength);

        foreach ($daysEvents as $event) {
            if ($event != null) {
                displayEvent($event, $kal);
            }
        }

        echo '</div>';
    }

function cmpEventLength($a, $b) {
return strlen($a['event_text']) - strlen($b['event_text']);
}

function popMultiDayEvent(&$daysEvents, $date) {
//    echo 'Days Events: ';
//    var_dump($daysEvents);

    $selectedLength = 0;
    $selectedIndex = null;

    for ($ct1 = 0; $ct1 < count($daysEvents); $ct1++) {
        $event = $daysEvents[$ct1];

        if (isMultiDay($event)) {
            $length = round( (strtotime($event['allday_end']) -
strtotime($event['allday_start'])) / 86400 );
            if ($event['allday_start'] == $date || $dayOfWeek == 0) {
                if ($length > $selectedLength) {
                    $selectedLength = $length;
                    $selectedIndex = $ct1;
                }
            }
        }
    }
}

```



```

        }
    }

}

if ($selectedIndex !== null) {
    $event = $daysEvents[$selectedIndex];
    $daysEvents = array_trim($daysEvents, $selectedIndex);
    return $event;
}
return null;
}

function array_trim( $array, $index ) {
    if ( is_array ( $array ) ) {
        unset ( $array[$index] );
        array_unshift ( $array, array_shift ( $array ) );
        return $array;
    } else {
        return false;
    }
}

function multiDayEventsLeft($weeksEvents, $dayOfWeek, $today) {
    for ($sct1 = 0; $sct1 <= $dayOfWeek; $sct1++) {
        $daysEvents = $weeksEvents[$sct1];
        for ($sct2 = 0; $sct2 < count($daysEvents); $sct2++) {
            $event = $daysEvents[$sct2];
            if (isMultiDay($event) && $event['allday_end'] > $today) {
                return true;
            }
        }
    }
    return false;
}

function isMultiDay($event) {
    return $event['allday_start'] != null && ($event['allday_end'] -
$event['allday_start'] > 1);
}

function displayEvent($event, $kal) {
    displayEventAux($event, true, true, $kal);
}

function displayEventAux($event, $start, $end, $kal) {
    global $master_array;
    $event_calno = 0;
    $event_calno = $event['calnumber'];
}

```

```

$event_text = htmlentities(stripslashes($event["event_text"]));
                //$event_calno = $event['calnumber'];
$eventDescription = $event['description'];
$eventDescription = str_replace("\n", "<br />", $eventDescription);
$eventDescription = str_replace("\r", "<br />", $eventDescription);
$eventDescription = htmlentities(stripslashes($eventDescription));
$uid = $event['uid'];
$calendars = $master_array['calendars'];
                //$calName = 0;
$calName = idEncode($calendars[$event_calno]['id']);
$allDay = "";

                $martti = NULL;
                $mitteri = $master_array['calendars'];
for ($sap = 1; $sap <= count($mitteri); $sap++) { //laskeskellaan kalenterit
                $calendar = $calendars[$sap];
                }

                //echo $sap;
                $kel = $sap-1;
                for ($kalenteri = 1;

$kalenteri < $sap; $kalenteri++) {

                $kel = $kalenteri;
                $kalla =

$mitteri[$kalenteri];

if (strstr($event_text,$kalla['name'])){$martti = $kel;}

                }

                if ($event_calno == $kal OR
preg_match("/Yhteinen/i", $event_text) OR
($martti !== NULL && $martti == $kal))
                {
                if ($event['allday_start'] != null) {
                        $allDay = ' all-day';
                }
                if ($start) {
                        $allDay .= ' start';
                }
                if ($end) {
                        $allDay .= ' end';
                }
                }

                $multiDayStartEnd = "";

                if (isMultiDay($event))
$multiDayStartEnd = ' start-date-' . $event['allday_start'] . ' end-date-' .
$event['allday_end'] ;
                }

```

```

$IdAttr = "";
if ($Suid) {
    $IdAttr = ' id="event-' . $Suid . "'";
}
echo '<div' . $IdAttr . ' class="event cal-' . $calName
    . ' color-' . $event_calno . ' ' . $allDay . $multiDayStartEnd . "'>';
echo '<div class="text-outer"><div class="text"'";
if ($eventDescription != null && false) {
    $popupHTML = '<div class=\\\\"title\\\\">' . addslashes($event_text) . '</div>'
        . '<div class=\\\\"body\\\\">' . addslashes($eventDescription) . '</div>';

    $popupHTML = '<div class=\\\\"body\\\\">' . addslashes($eventDescription)
        . '</div>';

    echo " onmouseover=\"domTT_activate(this, event, 'content', '"
        . $popupHTML . "', 'styleClass', 'niceTitle', 'fade', 'in', 'fadeMax', 87);\"";
}
echo '>&nbsp;';
if(stristr($event_text, 'tehty') != FALSE){
    echo '<s>';
    echo ($event_text == "") ? '&nbsp;': $event_text;
    echo '</s>';
}
else{
echo ($event_text == "") ? '&nbsp;': $event_text;}

if(stristr($event_text, 'yhteinen') == FALSE)
    {if($martti != NULL){echo " Siirretty "; echo $calName;
    echo ":lta";} }

echo '</div></div>';

echo '<div class="description">';
echo br2nl($eventDescription);
echo '</div>';

echo '</div>';
}
}

function br2nl($str) {
    return preg_replace('=<br */?>=i', "\n", $str);
}

function getEventLength($event, $dayOfWeek, $currentDay) {
    $length = round( (strtotime($event['allday_end']) - strtotime($event['allday_start'])) / 86400 )

```

```

$daysLeftInEvent = round( (strtotime($event['allday_end']) - strtotime($currentDay)) / 86400 );

$daysLeftInWeek = 7 - $dayOfWeek;

    return min($daysLeftInEvent, $daysLeftInWeek);
}

function eventsLeftInWeek($weeksEvents) {
    for ($sct1 = 0; $sct1 < 7; $sct1++) {
        $daysEvents = $weeksEvents[$sct1];
        if ($daysEvents !== null) {
            return true;
        }
    }
    return false;
}

function displayCalendarNames() {
    global $master_array;
    $calendars = $master_array['calendars'];

    echo '<div class="calendars" id="calendars">';
    for ($sct1 = 1; $sct1 <= count($calendars); $sct1++) {
        $calendar = $calendars[$sct1];
        $inputId = 'view-' . idEncode($calendar['id']);
        $isWebCal = "";
        if ($calendar['isWebCal']) {
            $isWebCal = ' webcal';
        }

        if (!defined(DEFAULT_CALENDAR)) {
            define(DEFAULT_CALENDAR, $calendar['name']);
        }

        $isDefault = "";

        if ($calendar['name'] == DEFAULT_CALENDAR)
            {
                $isDefault = ' default';
            }

    echo '<div class="calendar-info color-' . $sct1 . $isWebCal . $isDefault . "'>';
    echo '<span class="cal-details">';

    echo '<input type="checkbox" name="' . $inputId . '" id="' . $inputId . '"
checked="checked" />';
    echo '<label class="name" for="' . $inputId . "'>';
    echo $calendar['name'];
    echo '</label>';

```

```

echo '</span>';
echo '<span class="links">';

if ($calendar['isWebCal']) {
    echo '<a href="' . $calendar['filename'] . '">ics</a>';
    echo '<span class="faded">rss</span>';
    }

    else {
    echo '<a href="calendars/' . $calendar['id'] . '.ics' . '">ics</a>';
    echo '<a href="http://serveri/kalenteri/monket-cal-
source/phpicalendar-karl/rss/rss.php?cal=' . $calendar['id'] .
'&amp;rssview=month">rss</a>';
    }
    echo '</span>';

    echo '</div>';
    }
echo '</div>';
}

function idEncode($text) {
    $temp = preg_replace('/ +/', '-', $text);
    return preg_replace('/[^A-Za-z0-9-_:]/', '_', $temp);
}

function displayCalendar() {
    global $master_array;
    global $prev_month;
    global $next_month;
    global $week_start_day;
    global $daysofweek_lang;
    global $start_month_day;
    global $today_today;

    }

{
?>
<div id="baseURL" title="<?= SITE_DIR ?>" class="hidden"></div>
<div style="text-align: center; margin: 20px;"><a href="<?= SITE_DIR
?>">Nykyinen viikko</a></div>

<div id="calendar">

<?php echo '<div class="prev-month"><a href="index.php?apuri=' .
date("Ymd",strtotime("-1 week", $unix_time)) . '"></a>';?>
<?php echo '<a href="index.php?apuri=' . date("Ymd",strtotime("-1 week",
$unix_time)) . '"></a>';?>

```

```
<?php echo '<a href="index.php?apuri=' . date("Ymd",strtotime("-1 week",
$unix_time)) . "'></a>';?>
```

```
<?php echo '<a href="index.php?apuri=' . date("Ymd",strtotime("-1 week",
$unix_time)) . "'></a></div>';?>
```

```
<?php echo '<div class="next-month"><a href="index.php?apuri=' .
$first_of_month = $next_week . "'><&raquo;</a>'; ?>
```

```
<?php echo '<a href="index.php?apuri=' . date("Ymd",strtotime("+2 week",
$unix_time)) . "'><&raquo;</a>'; ?>
```

```
<?php echo '<a href="index.php?apuri=' . date("Ymd",strtotime("+3 week",
$unix_time)) . "'><&raquo;</a>'; ?>
```

```
<?php echo '<a href="index.php?apuri=' . date("Ymd",strtotime("+4 week",
$unix_time)) . "'><&raquo;</a></div>'; ?>
```

```
<a name="top"></a>
<div class="day-header">
<table cellspacing="0" class="header" summary="Days of Weeks (these act as
titles for follwing week tables)">
  <tr>
    <?php
      // loops through 7 times, starts with $week_start_day
      $start_day = strtotime($week_start_day);
      for ($i=0; $i<7; $i++) {
        $day_num = date("w", $start_day);
        $day = $daysofweek_lang[$day_num];
        echo '<th>' . $day . '</th>';
        $start_day = strtotime("+1 day", $start_day);
      }
    ?>
  </tr>
</table>
</div>
<?php
$kalenterit = $master_array['calendars'];
for ($sap = 1; $sap <= count($kalenterit); $sap++) { //laskeskellaan kalenterit
  $scalendar = $calendars[$sap];
}

$kal = $sap-1;
for ($skalenteri = 1; $skalenteri < $sap; $skalenteri++) {
  $scalnumber = $skalenteri;
  $skal = $skalenteri;
  $skalla = $kalenterit[$skalenteri];
  echo $skalla['name'];
```

```

$sunday          = strtotime("$start_month_day");
$whole_month    = TRUE;
$weekNum = 0;

do {
    {

        $weekNum++;
        echo '<div class="week">';
        echo '<table cellpadding="0" cellspacing="0" summary="Events for weeks
        commencing ' . date('jS F Y', $sunday) . "'>';
        echo '<tr>';

        for ($ct1 = 0; $ct1 < 7; $ct1++) {
            $thisDate = strtotime('+ ' . $ct1 . ' day', $sunday);
            $day = date("j", $thisDate);
            $daylink = date("Ymd", $thisDate);

            echo '<th>';
            if ($today_today == $daylink) {
                echo ' class="today"';
            }

            echo '>';
            if ($weekNum != 1) {
                echo '<div class="more show-full">&nbsp;</div>';
            }

            echo '<a class="day-number" href="' . $daylink . "'>' . $day .
            '</a>';
            if ($day == '1') {
                $month = date('F', $thisDate);
                echo '<span class="month-text">' . $month . '</span>';
            }

            echo '<a class="new-event" title="' . $daylink .
            "'><span>&nbsp; Uusi tieto</span></a>';
            echo '</th>';

            }

        echo '</tr>';

        // calculate max number of events in a single day this week
        $maxEventsPerDay = 0;
        $weeksEvents = array();
        for ($ct1 = 0; $ct1 < 7; $ct1++) {

```



```

    }
else {
echo '<td class="td-s" rowspan="" . (($maxEventsPerDay - $ct2)) . "">';
outputSingleDayEvents($daysEvents, $ct1, $daylink, $kal);
echo '</td>';
$weeksEvents[$ct1] = null;
    }
}

else {
    $eventLength = getEventLength($event, $ct1,
    $daylink);
    $start = ($daylink == $event['allday_start']);
    $daysLeftInEvent = round( (strtoime
    ($event['allday_end']) - strtotime($daylink))
    / 86400 );
    $end = ($daysLeftInEvent == $eventLength);

    $kal = $kalenteri;

    echo '<td class="td-m" colspan="" .
    $eventLength . "">';
    displayEventAux($event, $start, $end, $kal);
    echo '</td>';
    $ct1 += $eventLength - 1;

    }
}

}
echo '</tr>';
}

}
echo '<tr>';
for ($ct1 = 0; $ct1 < 7; $ct1++) {
    $thisDate = strtotime('+'. $ct1 . ' day', $sunday);
    displaySingleDayEvents($thisDate, $ct1);
}
echo '</tr>';
echo '<tr class="padding">';
for ($ct1 = 0; $ct1 < 7; $ct1++) {
    echo '<td>&nbsp;&nbsp;&nbsp;</td>';
}
echo '</tr>';

```

```

        echo '</table>';
        echo '</div>';

        $sunday = strtotime("+7 day", $sunday);
        $checkagain = date("m", $sunday);
        if ($checkagain != $this_month) $whole_month = FALSE;

    } while ($weekNum < 1); // while ($whole_month == TRUE);

    echo '<div class="dummy-week">';
    echo '<table cellpadding="0" cellspacing="0" summary="">';
    echo '<tr>';
    for ($sct1 = 0; $sct1 < 7; $sct1++) {
        echo '<td><div class="more show-full">&nbsp;</div></td>';
    }

    echo '</tr>';
    echo '</table>';
    echo '</div>';
}

?>

</div>

<div id="show-hide-full">
    <a href="#" class="show-full hidden">Näytä kaikki tapahtumat</a>
    <a href="#" class="hide-full hidden">Näytä vain tapahtumat jotka
mahtuvat</a>
</div>
<div class="clear"></div>

<?php
    displayCalendarNames();

?>

<?php
}

?>
<div style="text-align: center; margin-top: 85%;">
<?php
echo "<FORM action=\http://serveri/kalenteri/kuukausi/index.php\
method=\"post\">";

```

```

echo "<LABEL for=\"firstname\">Pvm esim: 20081201</LABEL>";
echo "<INPUT type=\"text\" id=\"paivamaara\" name=\"paivamaara\"><BR>";
echo "<input type=\"submit\" />";
echo "</form>";

```

```

?>
<div id="ylapalkisto">
Muokataan <?php if ($_POST["name"] != "") {echo $_POST["name"];} else {
echo "Yhteinen";} ?>. :n kalenteria

```

```

<?php
echo "<form action=\"index.php\" method=\"post\">
<select name=\"name\">";
$kalexit = $master_array['calendars'];
        for ($sax = 1; $sax <= count($kalexit); $sax++){
//laskeskellaan kalenterit
        $kalla = $kalexit[$sax];
        echo '<option value="" . $kalla['name'] . "'>' . $kal-
la['name'] . '</option>';

```

```

        }
echo "</select>";
echo "<input type=\"submit\" value=\"Valitse\"/>";
echo "</form>";
//Siirretään valittu kalenteriluokka muuttujajaan siirtoa varten
//echo "<center><a href=\"http://www.hsaoy.com\">
<img src=\"http://www.hsaoy.com/hsa/images/logohsa_w130.gif\" alt=\"HSA
OY\"></a></center>";
?>

```

```

</div>
<div id="valinta">
<?php
echo "<H2><br>";
echo date ("m / Y", $unix_time);
echo "</H2>";
//Siirretään valittu kalenteriluokka muuttujajaan siirtoa varten
?>

```

```

</div>
<div id="linkkilista">
<?php

```

```

echo "<center><a href=\"http://serveri/kalenteri/monket-cal-source/phpicalendar-
karl/print.php?cal=$tmpmuuttuja&getdate=$getdate&printview=day\">Tulostaa
valitun henkilön päiväkalenterimerkinnät</a></center>";
echo "<center><a href=\"http://serveri/kalenteri/monket-cal-source/phpicalendar-
karl/print.php?cal=$tmpmuuttuja&getdate=$getdate&printview=week\">Tulostaa
valitun henkilön viikkokalenterimerkinnät</a></center>";

```

```
echo "<center><a href=\"http://serveri/kalenteri/monket-cal-source/phpicalendar-
karl/print.php?cal=$tmpmuuttuja&getdate=$getdate&printview=month\">Tulosta
a valitun henkilön kuuukausikalenterimerkinnät</a></center>";
```

```
echo "<center><a href=\"http://serveri/kalenteri/monket-cal-source/phpicalendar-
karl/print.php?cal=$all_calendars_combined&getdate=$getdate&printview=day\"
>Tulosta kaikki kalenterit yhdessä päivänäkymä</a></center>";
```

```
echo "<center><a href=\"http://serveri/kalenteri/monket-cal-source/phpicalendar-
karl/print.php?cal=$all_calendars_combined&getdate=$getdate&printview=week\
\">Tulosta kaikki kalenterit yhdessä viikkonäkymä</a></center>";
```

```
echo "<center><a href=\"http://serveri/kalenteri/monket-cal-source/phpicalendar-
karl/print.php?cal=$all_calendars_combined&getdate=$getdate&printview=mont
h\">Tulosta kaikki kalenterit yhdessä kuukausinäkymä</a></center>";
```

```
?>
```

```
</div>
```

```
<?php
```

```
echo "<h3><center>X-ON calendar by: Mikko Vahter</center></h3>";
```

```
?>
```