

Simo-Matti Viljanen

INTERNETSOVELLUSKEHITYKSEN
OPETUS JA -OPETUSTYÖKALUJEN
VERTAILU

Opinnäytetyö
Tietojenkäsittely


Toukokuu 2010




MIKKELIN AMMATTIKORKEAKOULU

Mikkeli University of Applied Sciences

KUVAILULEHTI

 MIKKELIN AMMATTIKORKEAKOULU Mikkeli University of Applied Sciences		Opinnäytetyön päivämäärä 7.5.2010
Tekijä(t) Simo-Matti Viljanen	Koulutusohjelma ja suuntautuminen Tietojenkäsittelyn koulutusohjelma	
Nimeke Internetsovelluskehityksen opetus ja -työkalujen vertailu		
Tiivistelmä <p>Internetsovelluskehitysohjelmien määrä on suuri. Mikkelin ammattikorkeakoululla ei ole kahden viimeisimmän opetuskäytössä olleen ohjelman valinnassa ollut selkeitä kriteerejä. Tämän työn tehtävä on määrittää kriteerit ja suositella sopivaa opetusohjelmaa samalla tutustuen aiheen opetukseen.</p> <p>Opetettaessa internetsovelluskehitystä opetus tulee aloittaa helpoimmasta ja edetä vaikeimpaan samalla tukien opiskelijaa ongelmakohdissa. Opiskelijan tulisi saavuttaa tälle asetetut tavoitteet motivoituneena ja oma-aloitteisena.</p> <p>Vertailtavia ohjelmia on seitsemän. Notepad++, Programmer's Notepad, PSPad, PHP Coder, Emerald Editor, ConTEXT ja Komodo Edit. Ohjelmien taso ja ominaisuudet vaihtelevat, mutta PHP Coder:ia ja ConTEXT:ia lukuun ottamatta kaikki ohjelmat kävisivät opetukseen. PSPad ja Komodo Edit erottuvat joukosta parhaina kandidaatteina, sillä ne ovat kokonaisuutena kattavimmat ja opetuskäyttöön soveltuvimmat.</p>		
Asiasanat (avainsanat) tietojenkäsittely, atk-ohjelmat, opetus, kasvatustiede, vertailu,		
Sivumäärä 48s + liitteet 4s	Kieli Suomi	URN
Huomautus (huomautukset liitteistä)		
Ohjaavan opettajan nimi Arto Väättäinen	Opinnäytetyön toimeksiantaja Mikkelin ammattikorkeakoulu	

DESCRIPTION

 MIKKELIN AMMATTIKORKEAKOULU Mikkeli University of Applied Sciences		Date of the bachelor's thesis 7 May 2010	
Author(s) Simo-Matti Viljanen		Degree programme and option Business Information Technology	
Name of the bachelor's thesis The teaching of Internet application development and comparison of teaching tools			
Abstract <p>The number of Internet application development programs is huge. Mikkeli University of Applied Sciences had not had clear criteria for its teaching tools in Internet application development. The Assignment for this Bachelor's thesis was to define the bench marks and recommend the suitable tool for teaching. This bachelor's thesis also familiarizes how internet application development should be taught.</p> <p>The teaching of Internet application development should begin with easy tasks and proceed to the harder ones while in supporting students in problematic parts. Students should achieve their goals with motivated and initiative.</p> <p>There were seven programs to compare for coverage and suitability as a teaching tool. Those included Notepad++, Programmer's Notepad, PSPad, PHP Coder, Emerald Editor, ConTEXT and Komodo Edit. The professional level of those programs and with features varied. Except for PHP Coder and ConTEXT, all programs are suitable for training purposes. PSPad and Komodo Edit stood out as the best candidates as they were most comprehensive and the most suitable for the school's needs.</p>			
Subject headings, (keywords) Data proceeding, ADP programs, teaching, pedagogy, comparison.			
Pages 48 p. + appendix 4 p.	Language Finnish	URN	
Remarks, notes on appendices			
Tutor Arto Väätäinen		Bachelor's thesis assigned by Mikkeli University of Applied Sciences	

SISÄLTÖ

1	JOHDANTO	1
2	INTERNETSOVELLUSKEHITYKSEN OPETUS	2
2.1	Opetuksen suunnittelu	2
2.2	Opetus ja arviointi.....	3
2.3	Ohjelmien tehtävä.....	6
3	OHJELMIEN OMINAISUUDET.....	7
3.1	Notepad++	10
3.2	Programmer's Notepad.....	13
3.3	PSPad.....	16
3.4	PHP Coder	23
3.5	Emerald Editor.....	28
3.6	ConTEXT	34
3.7	Komodo Edit.....	36
4	OHJELMIEN VERTAILU	41
5	PÄÄTÄNTÖ	46
	LÄHTEET.....	47
	LIITTEET.....	49

1 JOHDANTO

Työn tavoitteena on etsiä Mikkelin ammattikorkeakoululle uusi internetsovelluskehitystyökalu ja tutkia samalla internetsovelluskehityksen opetusta. Koululla aikaisemmin käytössä ollut Crimson Editor korvattiin vuonna 2009 Notepad++-ohjelmalla. Tämän vaihdon pohjimmainen syy oli Crimson Editor:in vanheneminen ja vaihtaminen ajan mukaisempaan ohjelmaan. Nykyiseksi opetusohjelmaksi päätyi Notepad++, ilman sen kummempia vertailuja. Tavoitteena tässä opinnäytetyössä on toteuttaa tuo vertailu, jossa saatavilla olevia ohjelmia kilpailutetaan toisiaan vastaan ja katsotaan millä on parhaat edellytykset opetuksellisiin tarkoituksiin.

Tämä työ tutkii ohjelmia opetuksellisesta näkökulmasta. Kuinka opetuksen tulisi tapahtua ja mitä siinä pitää osata ottaa huomioon? Työssä otetaan kantaa internetsovelluskehityksen opetuksen suunnitteluun, toteutukseen, arviointiin ja siihen, millainen on hyvä opetusohjelma. Työ pyrkii myös arvioimaan, kuinka ohjelmat täyttävät niille asetetut kriteerit ja lopuksi antaa oman suosituksensa siitä, pitäisikö Notepad++ korvata, jollain vertailussa mukana olleella ohjelmalla.

Työssä käsiteltävien ohjelmien tulee täyttää seuraavat vaatimukset. Ohjelman tulee olla ilmainen ja helposti netistä imuroitavissa. Ohjelman tulee pystyä käsittelemään opetuksessa käytettäviä kieliä, joiksi tässä yhteydessä luetaan html, php, JavaScript ja css. Vertailtavien ohjelmien tulee olla soveltuvia sekä aloittelijoiden että edistyneempien opiskelijoiden opetukseen. Ohjelmien tulee myös nauttia suosituksesta ja olla kehitysjatkomossa. Kehitysjatkomolla pyritään laskemaan kynnystä omaksua ohjelmaa uusien tekniikoiden kehittyessä.

2 INTERNETSOVELLUSKEHITYKSEN OPETUS

Internet-sivut ovat viime vuosikymmenen aikana vakiintuneet kokonaisuudeksi useita eri tekniikoita. Itse koodi on pysynyt hyvinkin samana, vain tapa tehdä sitä on muuttunut. Joomlan, Drupalin ja muiden sivukokonaisuuksien rakentajien ansiosta, modernin nettisivun rakentaminen on helppoa ja nopeaa. Pohjana näille ohjelmille tarvitaan kuitenkin ymmärrystä siitä, kuinka sivustot rakentuvat ja miten niitä voi muokata haluamallaan tavalla. Tämän ymmärryksen sisäistäminen ja erilaisiin tekniikoihin tutustuminen on edellytys ammattimaiselle osaamiselle.

Seuraavassa osiossa tutkitaan internetsovelluskehityksen opetuksen suunnittelua, mitä pitää ottaa huomioon, ennen kuin opetusta edes aloitetaan, itse opetusta, ja opetusvälineenä käytettävän ohjelman osuutta opetuksessa.

2.1 Opetuksen suunnittelu

Tekniikka kehittyi nopeasti ja pakottaa etsimään opetuksen pohjaksi suhteellisen pysyviä yleistyksiä, joiden avulla muuttuva tekniikka on opittavissa ja ymmärrettävissä. Olen Koivulahti-Ojalan (2001, 11) kanssa samaa mieltä siitä, että vakiintuneet tekniikat eivät ole jääneet paikalleen vaan elävät yhä kehitysjatkuessa. Tekniikoiden yleistymisen ei tapahdu yhdessä yössä, joten siihen pystytään aina opetuksen kannalta vaurutumaan pitkällä aikavälillä ja arvioimaan uusien kielten ja ominaisuuksien opetusarvo. Kuten Koivulahti-Ojala (2001, 25) sanoo, opetuksen tulee sisältää vakiintuneet asiat siitä huolimatta, että tekniikka kehittyi. Tämä ei tarkoita, että opetuksen tulisi jauhaa paikallaan, vaan sen pitäisi pohjautua yleistyneisiin periaatteisiin ja jäsenyyksiin. Oppilaan tulee sisäistää ensisijaisesti tekniikka, ei niinkään opetustapaa tai ohjelmistokokonaisuuksia.

Koivulahti-Ojalan (2001, 115) mukaan materiaalin ollessa laadittu yleisten tavoitteiden pohjalta, tarvitsee päivittää vain käytännön osuus. Koivulahti-Ojala (2001, 59) muistuttaa myös, että opiskeltavien asioiden tulee olla ns. vaikeusjärjestyksessä eli myöhempien kurssien tulee pohjautua jo opiskeltuihin asioihin. Koivulahti-Ojalan

(2001, 61) mukaan kaikki oppimateriaalin hankinnasta opetussuunnitelmaan ja käytettäviin ohjelmistoihin on opettajan ratkaisuja.

Koko suunnittelun lähtökohdan Mervi Koivulahti-Ojalan ATK-opettajan oppaasta voi tiivistää lauseeseen: Tavoitteita pyritään saavuttamaan työskentelyn suunnitelmalla. Meisalo, Sutinen ja Tarhio (2003, 38) luettelevat erilaisia motivaatiotekijöitä ja mainitsevat mm. uteliaisuuden, menestymisen tarpeen, onnistumisen, välittömän palautteen ja palkkiot. Tavoitteiden täyttäminen motivaation keinoin loisi tilanteen, jossa oppilas saavuttaisi itsensä kannalta mielekkäitä tuloksia noudattaen samalla opetussuunnitelmaa.

2.2 Opetus ja arviointi

Tämä työ pohjautuu esimerkkiin, jossa on käytetty html-, css-, php- ja JavaScript-kieliä. Kaikki kielet on sijoiteltu omiin lohkoihinsa, jolloin ne voidaan jättää pois, muun koodin siitä pahemmin kärsimättä. Opetuksen kannalta esimerkki on huono, sillä asiat tulisi opettaa yksitellen osana omia kokonaisuuksia. Kuten Koivulahti-Ojala (2001, 55) toteaa, hyvällä suunnitelmalla asiat muodostavat kokonaisuuden, jota taustasta riippumaton opiskelija pystyy opiskelemaan. Tehtävien tulee olla mielekkäitä, motivoivia eivätkä ne saa olla liian hatusta tempaistuja. Tämän takia lähtökohtana tulisikin olla työelämän ongelmat taikka muut sovellustilanteet johon opiskelija voi, ja todennäköisesti tulee, törmäämään. Kuvan 1 esimerkki, sivulla 11, sopii loistavasti juuri sovellustilanteen vaatimukseen ja myöhemmin tässä luvussa mainittavan harjoitustyön aiheeksi vaativammilla kursseilla.

Kallin (2003, 70) mukaan oppilaille pitää antaa suoraa tukea opiskelun alussa ja vähentää sitä mukaa, kun oppilaan tietoisuus ongelmasta ja sen ratkaisusta kasvaa. Parhaimmillaan oppilas löytää itse ongelmia ja ratkaisuja niihin. Koivulahti-Ojalan (2001, 61) mukaan opettaja voi hyödyntää opetuksessa eri tapoja, esimerkiksi luentoja, demonstraatioita, ryhmätyötä, harjoitustyötä, projektityötä, itsenäisiä tehtäviä ja ohjelmoitua opetusta. Ohjelmoitua opetusta voidaan käyttää hyvin ohjelmoinnin opetuksessa. Ohjelmoinnissahan vaaditaan systemaattista ja loogista etenemistä. Opetusta voidaan käyttää harjoitteluun, kertaukseen tai uuden asian opetukseen. Menetelmän etuna

on se, että opiskelija saa välittömästi palautetta päätelmiensä oikeellisuudesta. Koivulahti-Ojalan (2001, 93) mielestä ohjelmoitu opetus soveltuu myös eriyttävään opetukseen, koska opiskelijat pystyvät etenemään omaa tahtiaan.

Packardin ja Racen (2003, 81) mukaan, tietokone tuo opiskelijalla mahdollisuuden tehdä itselleen soveltuvia tehtäviä ja kehittää itseään parhaaksi näkemällään nopeudella. Tietokonetta käytettäessä kuka tahansa voi hypätä vaativampiin osioihin ilman, että tarvitsee odottaa ketään toista. Tällaisia tapauksia varten tulisi opettajalla olla tarjota tehtävästä myös astetta vaikeampia versioita, esimerkiksi saman toiminnon käyttäminen toiseen tarkoitukseen. Nämä voivat ylittää, jopa kurssitavoitteet. Koivulahti-Ojala (2001, 87) vahvistaa tätä tapaa sanomalla, että opiskeltavan asian harjoittelun tapahtumisen tulisi olla mahdollisimman lähellä teoriaosuutta. Ei ole mahdoton ajatus, että harjoittelu ja opetus tapahtuisivat samaan aikaan. Kun suunniteltu kokonaisuus on päästy loppuun, on oppilaiden oppineisuus testattava laajemmalla tehtävällä, joka osoittaa heidän kykynsä itsenäiseen työskentelyyn.

Koivulahti-Ojalan (2001, 68) mukaan kannattaa valmiiksi varautua yleisimpiin virhetilanteisiin ja esitellä niitä ja niiden välttämistä oppilaille. Koivulahti-Ojala (2001, 69) kehottaa myös käymään ongelmia läpi opetuksen aikana niin yksilö tasolla kuin yhteisesti. Käytännössä tämä tarkoittaa globaalien ongelmien selvittämistä kaikille ja yksilöllisen ongelman henkilökohtaista läpikäyntiä.

Yllä mainittu kuvaus vastaa hyvin pitkälle ohjelmoinnin opetusta Mikkelin ammattikorkeakoulussa tällä hetkellä. Tekstissä mainittu laajempi tehtävä tarkoittaa käytännössä harjoitustyötä, jonka tulisi olla yhdistelmä kurssin aikana opituista ja käsitellyistä asioista, sisältäen myös edellisillä kursseilla opetettuja asioita. Harjoitustöiden aiheet ovat useimmiten vapaita, kunhan käsittelevät opetuksen kohteena olevia tekniikoita. Opiskelijan arviointia voidaan suorittaa harjoitustyön lisäksi myös portfolioilla, jota käsitellään tässä tekstissä myöhemmin.

Ohjelmoinnin opetus on muutakin kuin vain koodausta. Koivulahti-Ojalan (2001, 25) mukaan oppilaiden tulee hallita perustietojen ja -taitojen lisäksi alansa käsitteitä, termejä ja keskeisiä toimintaperiaatteita. Omaa alaa ja sen ongelmia, vastauksineen, tulee

sisäistää niin hyvin, että siitä pystyy käymään tarvittaessa keskustelua. Packardin ja Racen (2003, 135) mukaan englanninkielinen termistö on myös hyvä hallita. Tämä voidaan helposti sisäistää opetukseen vaihtamalla käytössä olevien ohjelmistojen kieli englanniksi.

Oppilaan tulee siis sisäistää myös alan termistöä, käytäntöjä ja malleja, vaikka tämä on enemmän opetuksellinen asia kuin ohjelmaan yhdistettä, on ohjelman tehtävä auttaa näiden asioiden sisäistämisessä. Yksi vaikuttava syy on myös, ettei kaikille termeille löydy hyvää suomenkielistä vastinetta. Viime kädessä pallo on kuitenkin opettajalla.

Oppilaan arvioinnin apuna käytetään Mikkelin ammattikorkeassa usein portfolioita. Tällä tarkoitetaan, tässä yhteydessä, annettujen tehtävien kokonaisuutta, joka on jaettu pienempiin osiin. Osat käsittävät yhden tai useita harjoitustehtäviä saman aiheeseen liittyen. Oikeaa tekotapaa ei useimmiten ole, vain oikea tulos. Harjoitustyö on usein osa portfolioita itseään.

Koivulahti-Ojalan (2001, 94 - 96) mukaan portfolioityöskentelyllä voidaan suunnata opiskelijaa aktiivisempaan ja itseohjautuvampaan suuntaan. Portfolioityöskentely ottaa huomioon eritasoiset opiskelijat, säädettävällä vaikeusasteellaan, ja auttaa oppilasta rajaamaan taitonsa ja kehitystarpeensa.

Tehtävien tulisi luoda oppilaalle saavuttamisen tunne. Kuten Taube (1998, 15) kirjassaan toteaa, työskentely on silloin tehokkaimmillaan kun oppilaalla on motivaatiota täyttää tälle asetettuja tavoitteita. Koivulahti-Ojala (2001, 86) taas toteaa, että tehtävät mahdollistavat harjoittelun ja tietojen soveltamisen. Tehtävät voivat myös motivoida oppilasta ja kehittää muita tälle asetettuja tavoitteita.

Käytännössä siis tehtävien pitäisi tukea sekä oppimista, että työelämässä vastaan tulevia haasteita. Tehtävissä tämä tarkoittaa yleisempien ongelmien kohtaamista. Portfolio toimii hyvin myös oppilaille, jotka eivät pysty sisäistämään jotain tiettyä tehtävää, se antaa heille mahdollisuuden keskittyä sen tarjoamiin muihin tehtäviin ja palata lopuksi vaikeammilta tuntuviin. Taube (1998, 16) toteaa myös että, itseluottamus, oppimishalu, kyky eritellä, arvioida omaa oppimistaan ja omien tavoitteiden asetus voi kasvaa

portfoliotyöskentelyn myötä. Arvostelun kannalta hyvin tehty portfolio selventää arvostelua, Tauben (1998, 24) Portfolio-kirjan mukaisesti: Portfolion tulee toimia mittarina siinä mitä oppilas on oppinut ja näin kuinka hän on saavuttanut tavoitteet, jotka hänelle on asetettu.

2.3 Ohjelmien tehtävä

Koko ajan kehittyvässä teknologiakierteessä ohjelmistot paranevat jatkuvasti ja tarjoavat erilaisia apuja ohjelman käyttämiseen. Samalla vapautuu myös paikka opetuksen helpottamiseen. Työkaluohjelmistojen tarkoitus on tässä yhteydessä luoda hyvä opiskelupohja, josta sekä untuvikko, että pitemmälle kehittynyt saavat toteutettua itseään ja saavuttavat halutut tulokset mahdollisimman helposti. Ohjelmien tulisi olla levinneitä ja kattavia.

Packardin ja Racen (2003, 78) mukaan olisi hyvä, jos opiskelijalle kerrotaisiin hänelle asetetuista tavoitteista ja kuinka ne pyritään täyttämään. Tästä johtuen opettajan tulisi hallita ohjelman käyttö kokonaisvaltaisesti. Pääasia on ohjelmoinnin opiskelulla, ei itse ohjelman, sillä nettisivustoa tehdessä tulee koodin olla ohjelmistoriippumatonta.

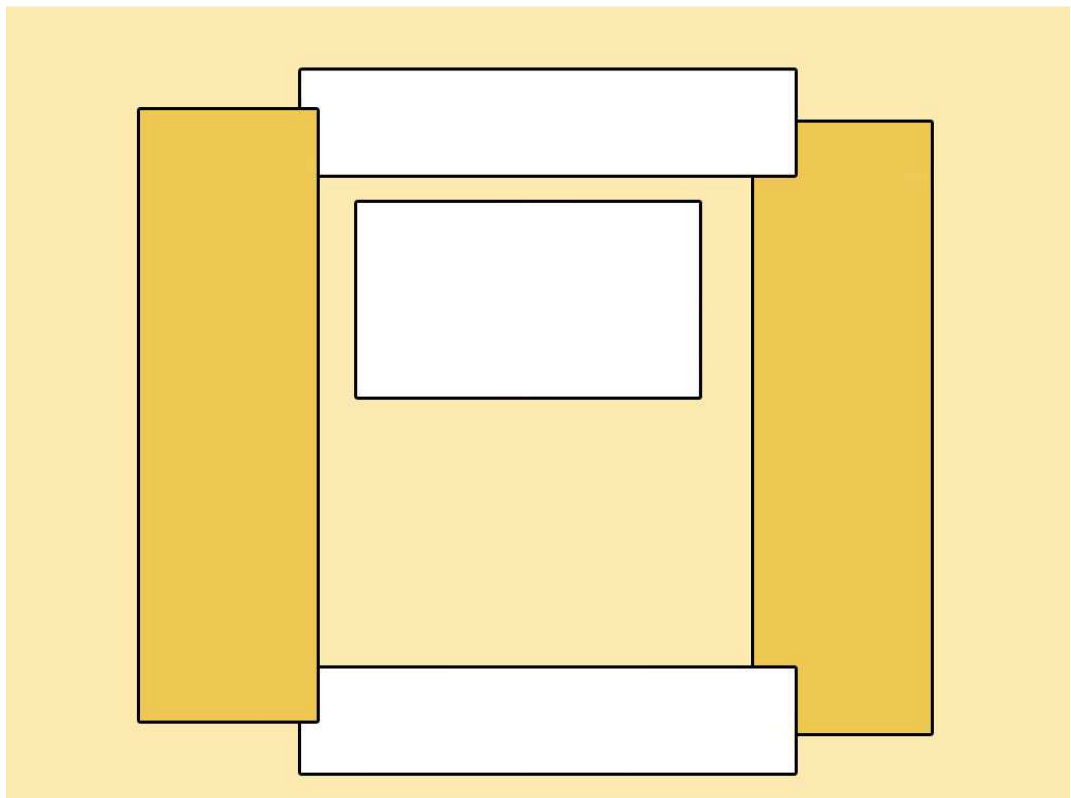
Koivulahti-Ojala (2001, 55) painottaa työvälineohjelmien laajaa kirjoa ja myöntää, että ohjelman toimintoja pitää rajata sen mukaan, mitä on järkevää opettaa. Tämän rajauksen teko on opettajan tehtävä. Rajat tulee asettaa niin, että oppilaat pystyvät ne määrällisesti sisäistämään. Jos ei ole tärkeä, kannattaa se jättää suosiolla pois. Tässä työssä onkin ohjelmien toiminnot rajattu niihin, joita oppilas todennäköisesti tarvitsee opiskellessaan tai työelämässä ollessaan.

Koivulahti-Ojala (2001, 67) määrittää avusteeksi ohjelman sisäiset toimintakokonaisuudet, jotka helpottavat ohjelman käyttöä. Oppilaan tulisi tietää mitä avusteita hänellä on mahdollisuus käyttää ja miten ne saadaan käyttökuntoon. Tämän työn yksi puoli on toimia ohjelmien avustuksena, eli oppaan kannalta tarvittavien ominaisuuksien käyttöohjeena.

3 OHJELMIEN OMINAISUUDET

Ohjelmien vertailu suoritetaan luomalla kuvan 1 mukainen sivusto, jossa on käytetty erilaisia tekniikoita. Sivusto on koottu käyttäen yksinkertaisia ja yleisiä elementtejä ja niiden ominaisuuksia. Näihin kuuluvat esimerkiksi elementtien taustaväri, reunaviiva ja asemointi. Sivuston lähdekoodi löytyy työnlopusta liitteenä. Tämä sivusto antaa mahdollisuuden ohjelman tärkeimpien toimintojen testaamiseen tilanteessa, jossa niitä voidaan tarvita. Sivuston pyörittämiseksi tarvitaan, php-kielen takia, paikallinen palvelin, jota tässä tapauksessa edustaa ohjelma nimeltä Wamp. Wamp on kokonaisvaltainen serveriratkaisu php:n pyörittämiseen sisältäen MySQL-tietokantaratkaisun. Sivustoa varten on luotu todella yksinkertainen MySQL-tietokanta, jossa on id, henkilön nimi ja henkilön puhelinnumero.

Mikä on paras editori ohjelmointitarkoitukseen? -lista ehdottaa php:n ohjelmointiohjelmaksi seuraavia: ConTEXT, UltraEdit, Visual Studio, .NET, PSPad, Notepad++ ja Crimson Editor. JEdit ja Komodo Edit mainitaan myös. Näistä vaatimuslistan ehdot täyttävät ConTEXT, PSPad, Notepad++, Crimson Editor ja Komodo Edit. (Mikä on paras editori ohjelmointitarkoitukseen? 2010.) Tommi Myrskykarin (2009) suosittelemana mukana on PHP Coder ja Mikro PC:n (2003) Programmer's Notepad.



KUVA 1. Toteutettavan sivuston graafinen asu

Jotkut ohjelmat on tarkoituksella jätetty pois vertailusta. Yksi näistä on Muistio, jota usein kuulee kotisivujen teko-ohjelmaksi ehdotettavan. Muistio ei kuitenkaan sisällä yhtään arvioimisen arvoisia ominaisuuksia, suomenkielen lisäksi. JEdit on jätetty pois sen vanhanaikaisen käyttöliittymän takia. Microsoftin Visual Studio olisi kyllä pärjännyt vertailussa, mutta se on liian massiivinen ratkaisu ongelmaan ja tässä vertailussa ei haeta Visual Studion tapaista jättiläisratkaisua. Toisin sanoen Visual Studion on liian hyvä.

Koska jokainen ohjelma on erilainen ja ohjelmat sisältävät paljon säädettäviä asioita, opastaa tämä työ vain oleellisimpien asetusten läpi. Näitä asetuksia ovat: automaattisten päivitysten poistaminen käytöstä, koska koulun päivityspolitiikka päivittää ohjelmistot tietyin väliajoin, eivätkä sitä vauhtia kuin ohjelmistokehittäjät niitä kehittävät. Opetuksen seuraamisen helpottamiseksi rivinumerot ovat oleellisia, joten ne otetaan näkyviin. Koska keinoja tähän voi olla useita, kerrotaan näistä tavoista yksi. Ohjelmoinnin nopeuttamiseksi selvitetään, mistä tekstin automaattisen täydennyksen ja -tekstinsyötön saa päälle ja minkälaisia alkuvalmisteluja se tarvitsee. Tähän sisältyy html-tagien, tyyliohjeiden ja php-komentojen automaattinen ennustus ja täydennys, sekä ” ja ’-merkkien ja erilaisten sulkujen automaattinen täydennys. Kielen muuttaminen mainitaan, jos ohjelma tukee suomen kieltä, vaikka tämä ominaisuus ei ole käytössä opetuksen aikana, se voi auttaa etsiessä termeille suomenkielistä vastinetta. FTP-tiedostojen siirto-ominaisuudelle ei koulun ympäristössä ole juuri käyttöä, mutta ammattiympäristössä toiminto on tärkeä. Lopuksi tutkitaan jokaisen ohjelman erikoisominaisuuksia ja toimintoja, jotka ohjelma tekee huomattavasti paremmin kuin muut tai ominaisuus on ohjelmalle uniikki. Varsinkin asennuksia muuttaessa käydään asiat siinä järjestyksessä läpi kuin ne ohjelman valikossa tulevat eteen.

Lyhyesti siis ohjelmien testattavien asioiden lista on seuraava:

- automaattisten päivitysten poisto
- rivinumerot
- automaattisen tekstinsyötön päälle laittaminen ja asetukset
- kielenvaihto suomeksi
- FTP-yhteyden luonti
- erikoisominaisuudet.

Ohjelmilla on useita samoja vähemmän tärkeitä ominaisuuksia. Nämä esitellään ensimmäisessä ohjelmassa, jossa ne tulevat vastaan. Usein ominaisuuden toiminta on lähes sama jokaisessa ohjelmassa, siksi sen toimintaa ei esitellä kuin kerran.

Sivuston kaikki asemoinnit ja värit toteutetaan tyylien määrityksillä, käyttäen sivun pohjana html-kieltä, johon ominaisuudet on upotettu. Ominaisuuksia tyylien lisäksi ovat JavaScript ja php-kieli. JavaScriptin avulla tarkistetaan onko lomakkeeseen syötetty tekstiä. Jos lomake on tyhjä, niin käyttäjälle ilmoitetaan siitä. Jos teksti on syötetty, ei JavaScript tee mitään. Php:n tehtävä on ottaa MySQL tietokantaan yhteys, hakea sieltä tietoja ja tulostaa ne taulukkoon.

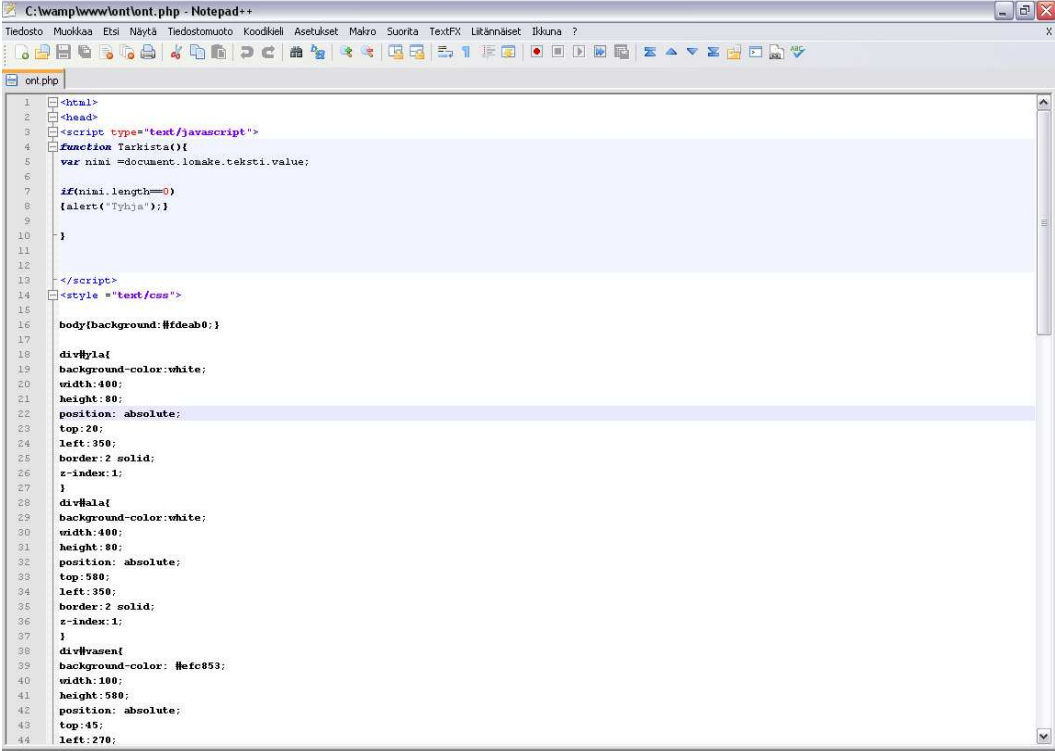
Tekstissä puhuttaessa hiiren oikeasta, tarkoitetaan valikkoa, joka saadaan esille painamalla hiiren oikeaa näppäintä kohteen päällä. Tyylietiedoilla tekstissä tarkoitetaan html-koodin elementtien tyylihallintakieltä virallisesti nimeltään Cascading Style Sheet (CSS), yleisesti puhutaan nettisivun tyylistä tai tyylietiedostosta, jos tyyli on eroteltu omaksi tiedostokseen. Puhuttaessa laajennuksista tarkoitetaan Plugin-toimintoja, joiden avulla ohjelman ominaisuuksia voidaan lisätä tai laajentaa entisestään.

Automaattisella tekstinsyötöllä ohjelmassa tarkoitetaan ohjelman ominaisuutta täydentää käyttäjän kirjoittamaa tekstiä ja arvata funktiota tai ominaisuutta. Tämän avulla käyttäjä vapautuu pitkien termien oikeinkirjoitusongelmasta.

3.1 Notepad++

Notepad++ on avoimeen lähdekoodin perustuva, C++-kielellä kirjoitettu, ohjelma. Ohjelmaa on kehitetty vuodesta 2003 tähän päivään ja sinä aikana se on voittanut kaksi kertaa Community Choice Award for Best Developer Tool-palkinnon. (Notepad++, 2010)

Kun Notepad++ -ohjelman käynnistää ensimmäisen kerran, etsii se netistä automaattisesti päivityksiä ohjelmalle itselleen ja sen laajennuksille sekä ilmoittaa, jos näitä on saatavilla. Laajennusten päivittäminen vaatii ohjelman uudelleen käynnistämisen ja ohjelman päivittäminen uudelleen asennuksen. Päivitysten automaattisen etsimisen saa pois päältä painamalla Plugins-valikosta Plugin manager-lehdeltä kohtaa Show plugin manager. Plugin manager-valikosta Settings-asetuksesta löytyy kohta Notify of plugin updates at startup. Tämän kohdan valitsematta jättäminen poistaa laajennuksien tarkastamisen käytöstä. Itse ohjelman päivittämisen saa pois päältä valitsemalla Settings-valikosta Preferences. Preferences-valikosta kohdasta MISC. löytyy valinta Enable NotePad++ auto-updater. Tämän toiminnon poistaminen käytöstä estää, ettei ohjelma enää yritä etsiä tietoa uudemmasta versiosta. Samasta-Preferences valikosta saa



```

C:\wamp\www\ont\ont.php - Notepad++
Tiedosto Muokkaa Etsi Näytä Tiedostomuoto Koodikeli Asetukset Malro Suorita TextFX Liitännäiset Ikkuna ?
ont.php
1 <html>
2 <head>
3 <script type="text/javascript">
4 function Tarkista(){
5 var nimi =document.lomake.teksti.value;
6
7 if(nimi.length==0)
8 {alert("Tyhjää");}
9
10 }
11
12
13 </script>
14 <style ="text/css">
15
16 body{background: #fdeab0;}
17
18 div#l1a{
19 background-color:white;
20 width:400;
21 height:80;
22 position: absolute;
23 top:20;
24 left:350;
25 border:2 solid;
26 z-index:1;
27 }
28 div#l1a{
29 background-color:white;
30 width:400;
31 height:80;
32 position: absolute;
33 top:580;
34 left:350;
35 border:2 solid;
36 z-index:1;
37 }
38 div#vanen{
39 background-color: #efc853;
40 width:180;
41 height:580;
42 position: absolute;
43 top:45;
44 left:270;

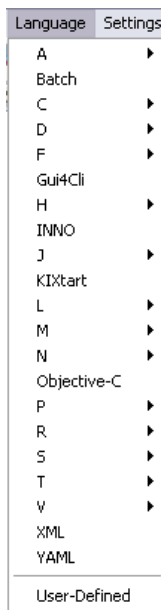
```

PHP Hypertext Preprocessor file 1640 chars 1874 bytes 118 lines Ln : 22 Col : 20 Sel : 0 (0 bytes) in 0 ranges Dos/Windows ANSI OVR

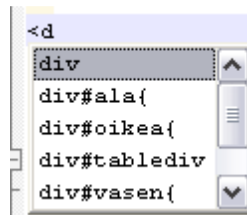
KUVA 2. Notepad++ ohjelman yleiskuva



KUVA 3. Notepad++ ohjelman painikkeet.



KUVA 5. Kielivalikko.



KUVA 4. Esimerkki ennustavasta tekstinsyötöstä.

Editing-kohdasta rivinumerot näkyviin valitsemalla Display line Numbers. General-kohdasta pystyy muuttamaan käyttöliittymän kielen suomeksi.

Kuvassa 3 on nähtävissä ohjelman erikoisnäppäimiä. Kuvan ensimmäisestä ja toisesta suurennuslaseista painaessa teksti suurenee ja pienenee. Neljänneksi viimeinen, Light Explorer, avaa tiedostoselaimen.

Automaattisen tekstinsyötön saa päällä Settings-valikosta valitsemalla Preferences. Tästä valikosta kohdasta Backup/Auto-Completion löytyy Enable auto-completion-valinta, jolla ennustuksen saa päälle tai pois. Samaisesta valikosta on suositeltavaa valita vaihtoehto word complication, jolloin ohjelma ennustaa tekstiä sanojen perusteella ja ottaa mukaan mm. php-muuttujat sekä tyyli luokat kuten kuvasta 4 voi nähdä. Tuki sanoille Else ja Ifelse puuttuu ennustuksesta, mutta värikoodaus näiden osalta toimii. Ohjelma ei myöskään tue lainausmerkkien tai sulkujen automaattista täydennystä. Päätteet värittyvät molemmista päistä kun kohdistin osuu jompaankumpaan. Ohjelma tunnistaa sanat ja osaa värittää saman sanan koko dokumentissa. Poikkeuksena aloittavat ja lopettavat tagit.

Jotta ohjelmassa saisi varit nakyyviin uudessa tiedostossa, tulee tiedosto tallentaa tai valita kuvan 5 valikosta haluttu syntaksikieli. Samalla ohjelma ottaa kyttoon ennustavan tekstinsyoton kyseiselle kielelle. Kielentunnistukset ovat kuitenkin yksilollisia eika esimerkiksi php-kielen tila ymmarra html-koodausta, jolloin kieli pitaa vaihtaa valikosta aina tilanteen mukaan.

```

1  <html>
2  <head>
3  <style type="text/css">
61 </head>
62 <body>
63   <div id="yla"> </div>
64   <div id="ala"> </div>
65   <div id="vasen"> </div>
66   <div id="oikea"> </div>
67   <div id="tablediv">
68   <table border="1" width="100%">
69   <tr>
70   <td></td><td></td>
71   </tr>
72
--

```

KUVA 6 Esimerkki koodin piilottamisesta.

Notepad ++ mahdollistaa tilaa vievan koodin piilottamisen. Piilottaminen tulee kyttetvaksi kun tagin jalkeen painaa Enter-nappainta, jolloin tagin ylosa toimii otsikkona. Piilottaminen onnistuu koodin vasemmalla puolella olevasta miinusmerkista. Kuvassa 6 on piilotettu sivun tyyliasetukset. Piilotetun osion huomaa plusmerkista, katkaisevas- ta viivasta ja rivinumeroiden hyppaamisesta. Piilotukset ovat kielisidonnaisia eli kie- lenvaihtaminen nolaa piilotukset. Piilottaminen ei vaikuta Kopioi-toiminnon toimin- taan. Esimerkiksi Valitse kaikki ja Kopioi-yhdistelma kopioi myos piilotetun aineiston.

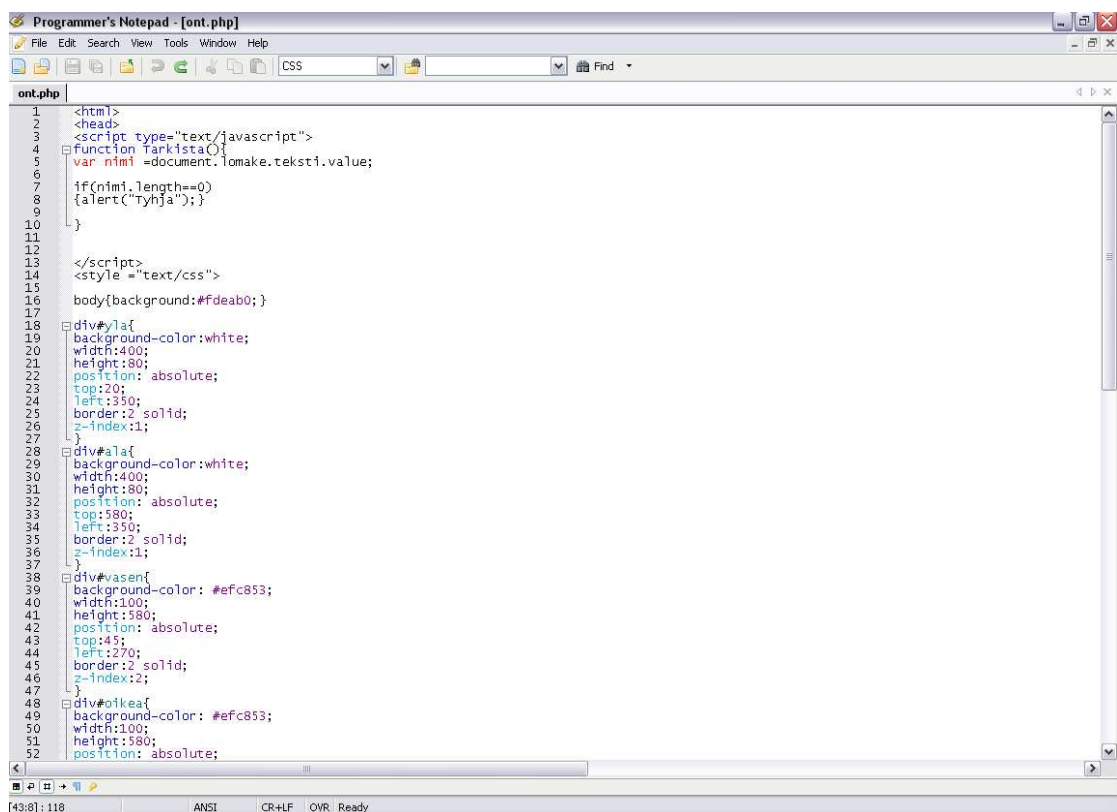
File-valikon Load Session... ja Save Session... -toiminnoilla voidaan luoda projekti- maisia kokonaisuuksia. Save Session... toiminto tallentaa paatteettomaan projektitie- doston, joka sisaltaa kaikki silla hetkella auki olevat tiedostot. Load Session... taas avaa taman, sulkien kaikki muut auki olevat tiedostot.

3.2 Programmer's Notepad

Programmer's Notepad on Simon Steelen, C++-kielellä, kirjoittama tekstieditori. Kuten kuvasta 7 voi todeta, ohjelma sisältää värikoodauksen ja välilehdet. (Programmer's Notepad, 2010)

Ensimmäisen kerran ohjelmaa avattaessa ilmoittaa testikoneen palomuuuri sen pyrkimyksestä päästä nettiin. Tämän ehkäisemiseksi otamme päivitysten tarkistamisen pois käytöstä. Tools-valikosta valitaan Options, jolloin aukeaa samanniminen valikko. General-kohdasta löytyy Check for updates, tästä poistetaan valinta. Kohdasta Visual Help valitaan kohta Highlight all instances of selected word, jolloin maalatun sanan värittäminen koko dokumentissa saadaan toimimaan. AutoComplete-valikosta tulee valita kaikki vaihtoehdot ja AutoComplete After-kenttään kirjoittaa pieni luku, testatessa käytetään lukua 1, jolloin tekstin ennustus alkaa toimia, kun on kirjoitettu sanan ensimmäinen merkki.

View-valikosta rakensittamalla kohdan Line Numbers, saa esiin rivinumerot. Saman valikon Windows-lehdeltä rakensittamalla kohdan File Browser, saa esiin hakemistopuun.

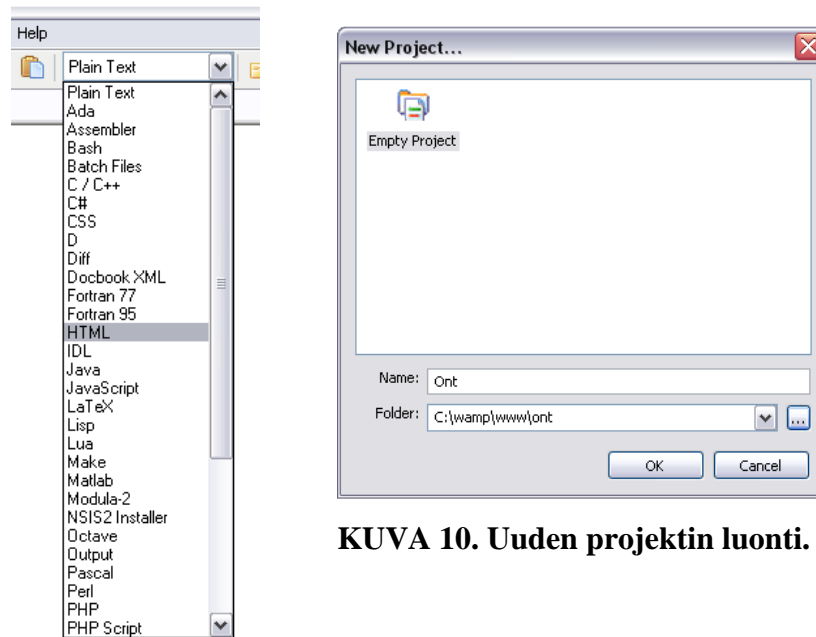


```

Programmer's Notepad - [ont.php]
File Edit Search View Tools Window Help
CSS Find
ont.php
1 <html>
2 <head>
3 <script type="text/javascript">
4 function Tarkista()
5 {
6   var nimi=document.lomake.teksti.value;
7   if(nimi.length==0)
8     {alert("tyhjä");}
9   }
10 }
11
12
13 </script>
14 <style ="text/css">
15
16 body{background:#fdeab0;}
17
18 div#y1af{
19   background-color:white;
20   width:400;
21   height:80;
22   position: absolute;
23   top:20;
24   left:350;
25   border:2 solid;
26   z-index:1;
27 }
28 div#a1af{
29   background-color:white;
30   width:400;
31   height:80;
32   position: absolute;
33   top:580;
34   left:350;
35   border:2 solid;
36   z-index:1;
37 }
38 div#vasenf{
39   background-color: #efc853;
40   width:100;
41   height:580;
42   position: absolute;
43   top:45;
44   left:270;
45   border:2 solid;
46   z-index:2;
47 }
48 div#oikeaf{
49   background-color: #efc853;
50   width:100;
51   height:580;
52   position: absolute;

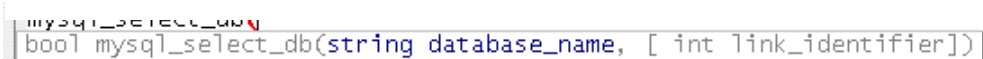
```

KUVA 7. Programmer's Notepad ohjelman yleiskuva.



KUVA 10. Uuden projektin luonti.

**KUVA 8. Kielen-
vaihtaminen Prog-
rammer's Note-
pad:ssa.**



KUVA 9. Esimerkki tekstillisestä avusta.

Automaattinen täydennys ei toimi suoraan asetuksen jälkeen testattavassa versiossa vaan vaatii <http://code.google.com/p/pnotepad/issues/detail?id=578> -osoitteesta ladattavan apis.zip-paketin ja sen sisällön siirtämisen asennushakemiston schemes-kansioon (esim. C:\Program Files\Programmer's Notepad\schemes). Ohjelmassa on automaattinen tagiparien täyttö. Kun avaavan tagin loppuun kirjoittaa suurempi kuin merkin, saa tagille sulkevan parin.

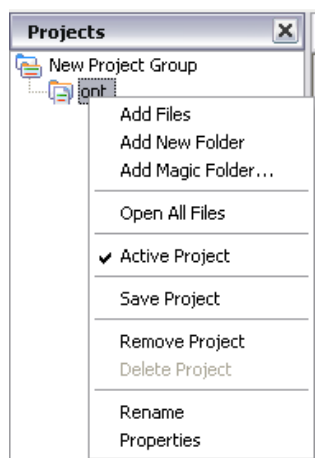
Ohjelman värikoodaus pohjautuu käyttäjän valintoihin. Työkalurivistä löytyvästä pudotusvalikosta voi valita halutun kielen. Kuten kuvasta 8 näkee, kielten määrä on suuri.

Ohjelma osaa etsiä sanoja dokumentista maalauksen perusteella, tämä ei eroa juuri mitenkään Notepad ++-ohjelman samaisesta ominaisuudesta. Automaattinen täydennys vaihtelee valitun kielen mukaan. Tallennus palauttaa kielen oletuskieleksi, eli siksi jolla tiedosto on tallennettu. Tässä yhteydessä php:ksi. Usein tallentaessa tämä turhauttaa.

Kirjoitettaessa MySQL-komentolausekkeen ensimmäistä sulkua tarjoaa ohjelma kuvan 9 mukaisen opastuksen sille. Tämä opastus ei tule tavanomaisimmilla komennoilla, kuten while tai for.

Projektivalion saa esille ottamalla View-valikosta Windows-lehdeltä Projects-valinnan. Tällöin vasempaan laitaan aukeaa projekti-ikkuna. Ikkuna aukeaa myös luotaessa uutta projektia New-valikon Project-painikkeella. Tässä valikossa valitaan Empty Project, annetaan sille nimi ja paikka, kuten kuvassa 10. Klikkaamalla hiiren oikealla luodun projektin päällä saadaan näkyviin kuvan 11 valikko. Add Magic Folder.. – lisää projektin rakenteeseen kansion, joka näyttää jo olemassa olevan kansion tiedostoja. Näytettävien tiedostojen tunnisteita voi myös säätää luontiprosessin aikana, jolloin näkyviin tulee esimerkiksi vain html- ja/tai php-päätteiset tiedostot.

Projektikokonaisuuksien hallinnassa ohjelmassa näyttäisi olevan bugi. Projekti ryhmää, ohjelmassa nimellä Project Group, tallentaessa pitää muistaa laittaa ryhmän nimen perään päätte .ppg, muuten ohjelma tallentaa ryhmään päätteettömästi, eikä osaa lukea sitä projektikokonaisuutena. Projektikokonaisuudet ovat usean projektin muodostama ryhmä.



KUVA 11. Projektin ominaisuuksia.

3.3 PSPad

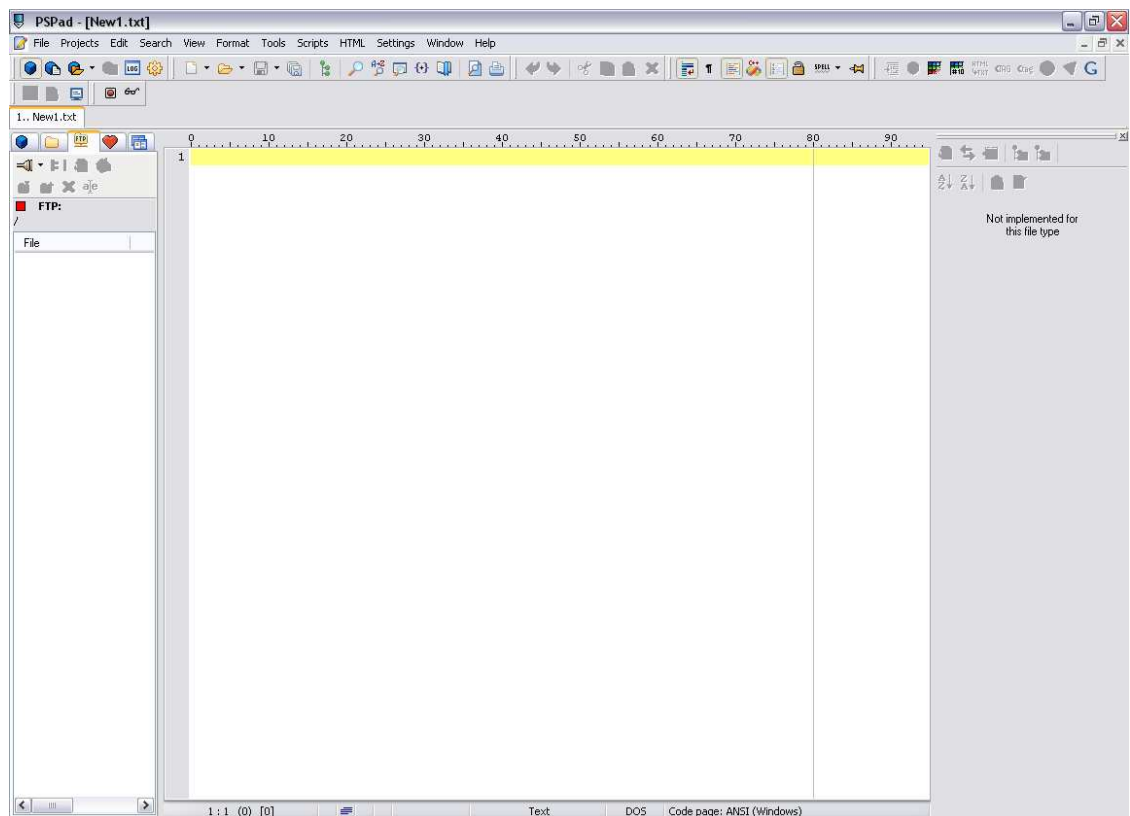
Jan Fiala julkaisi PSPadin ensimmäisen kerran vuonna 2001. Suurimmat ohjelmistojen jako-sivustot ovat kehuneet PSPadia sen laajoista ominaisuuksista ja ilmaisuudesta. (PSPad, 2010)

Kuten aikaisemmatkin ohjelmat haluaa PSPadkin yhteyden verkkoon. Kuitenkaan valikoista ei löydy tälle asetusta. Oletan, että ohjelma etsii tietoa uudemmasta versiosta. Kirjoitushetkellä seuraava beetaversio oli saatavilla, mutta ohjelma ei siitä noteerannut.

Rivinumeroit saa näkyviin View-valikosta valitsemalla Line Numbers.

Settings-valikon Program Settings-alavalikossa tulee ensimmäisenä eteen Language-vaihtoehto, josta pystyy vaihtamaan ohjelman käyttöliittymän kielen.

Editor(part 1) -valikosta rasti ruudussa ”Completion of chars ({{[<” ” tuottaa em. merkeille automaattisesti parin tekstiin, kun merkin edellä ei ole toista merkkiä.



KUVA 12. PSPad-ohjelman yleiskuva.

```

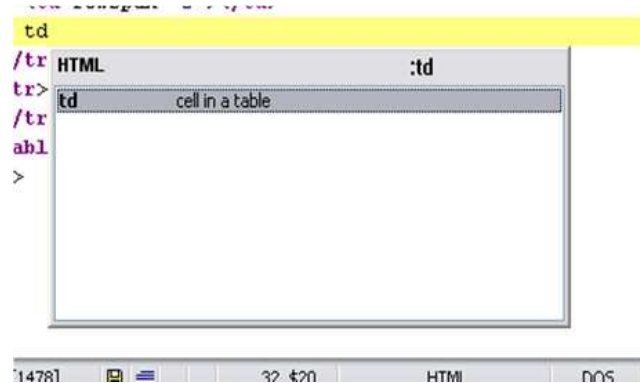
while () {
}

if () {
}

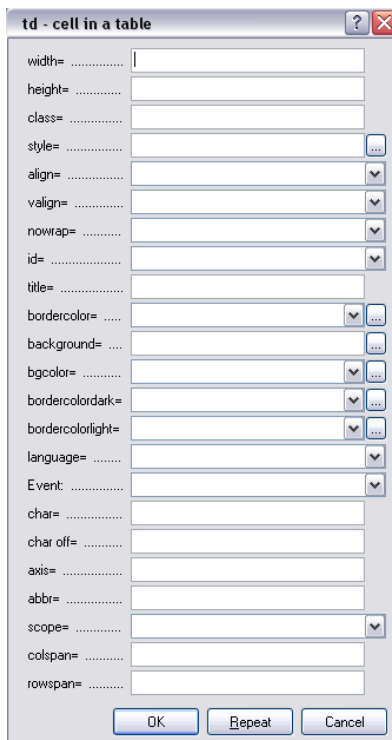
for (; ; ) {
}

```

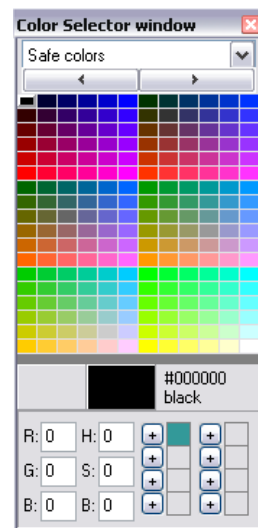
KUVA 13. Esimerkki sulkujen täydennyksestä.



KUVA 14. PSPad tunnistaa td-tagin.



KUVA 15. Td-tagin attribuuttivalikko.



KUVA 16. Värivalikko.

Program(part2)-valikosta voi myös halutessaan ottaa Show Splash Screen-valinnan pois päältä.

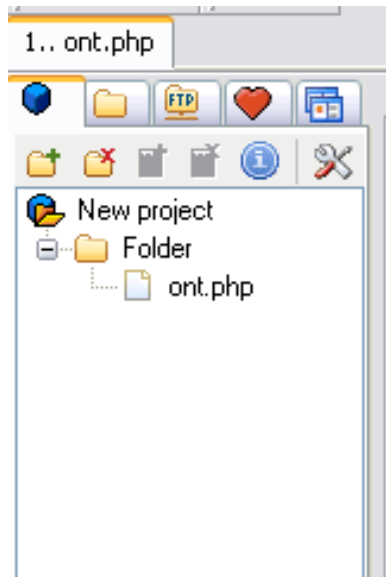
Automaattinen täydennys-lista löytyy, kun sanaa kirjoittaessa painaa Ctrl- ja J-kirjainta tai Ctrl ja välilyöntinäppäintä, erona näille vaihtoehdoille on, että jälkimmäisen mukana tulee funktion tarvitsemat lisämerkit kuten sulut tai on yhtä kuin-merkki, kuva 13 demonstroi tätä while-, if- ja for-funktioilla. Yksi PSPadin ydinominaisuuksista on html-elementtien nopea kasaan parsiminen, joka toimii myös Ctrl ja välilyönti



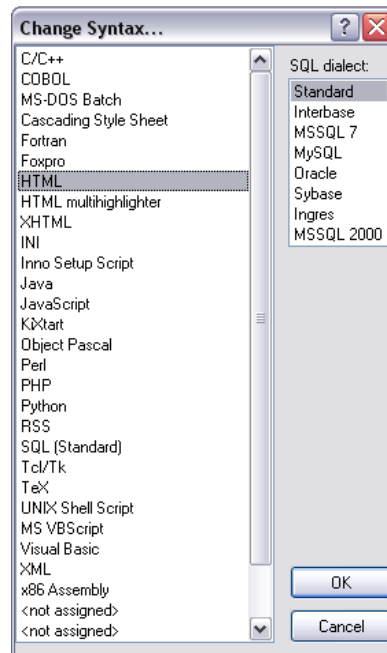
KUVA 17. PSPadin painikkeita.



KUVA 18. PSPadin erikoispainikkeita.



KUVA 19. Projektienhallintavälilehti PSPadissa.



KUVA 20. Värisyntaksin vaihtovalikko PSPadissa.

yhdistelmällä. Tagia kirjoitettaessa esiin tulee lista valittavana olevista tageista, kuva 14 demonstroi taulukon solun luontia, ja joko hiirellä tai Enterin painalluksella pääsee kuvan 15 valikkoon määrittelemään mahdollisia attribuutteja.

Repeat-painikkeella kuvassa 15 on sama vaikutus kuin OK-painikkeella, mutta se ei sulje attribuutti-ikkunaa ja vaan tyhjentää sen ja siirtää kohdistimen juuri luotujen tagien, tässä tapauksessa `<td>` ja `</td>`, väliin. Jos attribuuttini liittyy väreihin, osaa PSPad tarjota siihen avuksi kuvan 16 valikon. Kuvan 15 valikon attribuuttitermistö muuttuu aina valitun kielen mukaan ja ainakin tyylitiedot on todella nopea kirjoittaa. Kirjoittaessa määritystä diville, jonka Id on ylä, ei diville saa kaikkia määrittäyksiä kerralla, mutta jokaiselle attribuutille kyllä löytyy vastaava taulukko kuin td-tagille kuvassa 15.

Kuvan 17 neljännestä vasemmalta, Visual Studion logoa muistuttavasta symbolista, jonka päällä lukee C++, saa näkyviin Change Syntax-valikon. Valikosta voi vaihtaa tilanteen mukaan halutun korostuksen koodille.

Kuvan 18 ensimmäinen nappi sientää HTML-tekstin helppolukuisemmaksi. Tämä toiminto ei saata toimia tyylytietojen kanssa, vaan tiivistää ne yhdeksi kimpuksi. Asiaa voi yrittää korjata painamalla HTML-valikosta Reformat into Structured CSS-kohtaa, joka taas saattaa sotkea html-koodin asettelun. Paras tapa saada molemmat asemoitua on maalata käsiteltävä koodi ja tehdä sisennystoiminto sen jälkeen.

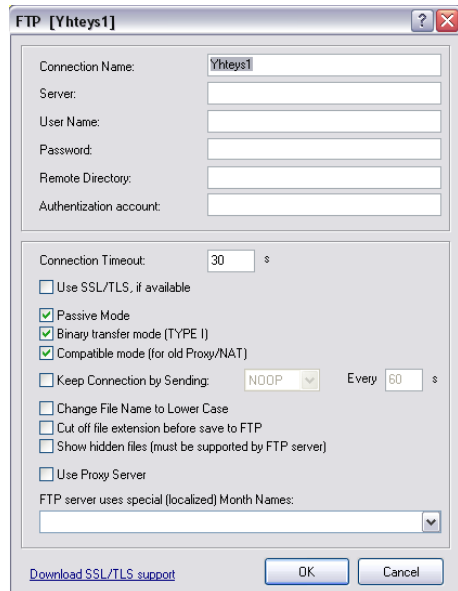
Kuvan 18 toinen nappi, oranssi laatikko, tiivistää kaiken tekstin mitä tiedostossa on, käytännössä tämä poistaa ”ilman” välistä. Tämä ja ensimmäinen nappi vaativat, että tiedosto on tallennettu.

Kolmas ja neljäs näistä tuovat esiin erilaiset värikartat, joista on helppo tutkia haluttuja värejä ja niiden värikoodeja. Viides luo uuden txt-välilehden, jossa on sivun teksti ilman tageja, tämän toiminta on vajaa ja tulos voi olla muu kuin haluttu. Kuudes ja seitsemäs muokkaavat kaikki tagit, joko isoiksi tai pieniksi kirjaimiksi. Kahdeksas on esikatselu html-muodossa. Php ei tässä muodossa toimi, mutta tyylit ja JavaScript toimivat.

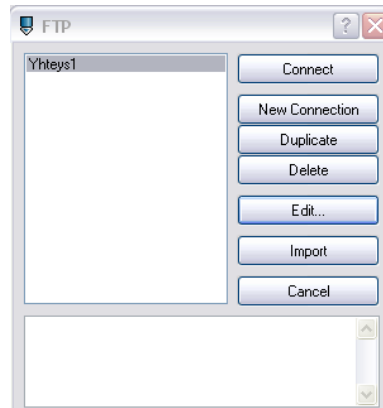
Jos työkalupaneeli eli ole näkyvässä ohjelman vasemmassa reunassa saa sen auki View-valikosta valitsemalla Tool Panel. Samalla toiminnolla paneelin voi myös sulkea, jos sitä ei tarvitse. Kuvasta 19 voi nähdä, että paneeli sisältää viisi välilehteä. Ensimmäinen vasemmalta on projektien hallintaan ja luontiin, toinen näyttää tiedostoselaimen ja kolmas on FTP-yhteyttä varten. Neljäs on paikka koota itselleen suosikkilistoja ja viides näyttää auki olevat tiedostot. Näistä toiminnoista keskitymme seuraavaksi oleellisimpiin eli projektien käsittelyyn ja FTP-yhteyteen.

Jotta projektiin voi lisätä tiedostoja, tulee New Project-hakemistoon luoda kansio. Lisääminen onnistuu joko raahaamalla auki olevan tiedoston välilehden kansion päälle tai valitsemalla, joko Project-valikosta tai hiiren oikealla kansion päällä, Add Files...

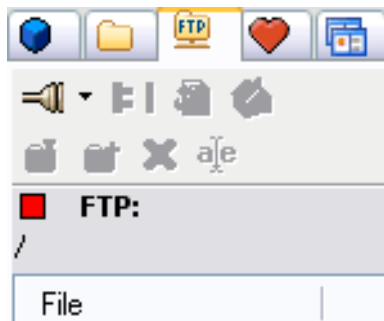
tai Add all open files. Huomaa, että projektien tallennus hoituu File-valikon vieressä olevasta Projects-valikosta.



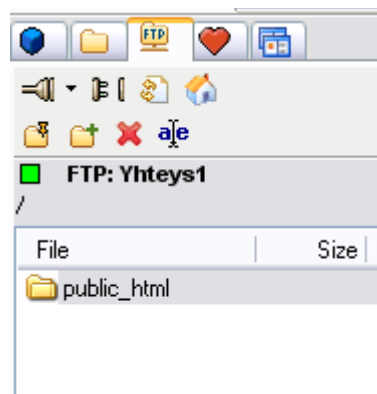
KUVA 21. FTP-yhteyden määrittäminen.



KUVA 22. FTP-yhteysvalikko.



KUVA 23. FTP-paneeli ilman yhteyttä.



KUVA 24. FTP-paneeli yhteydellä.

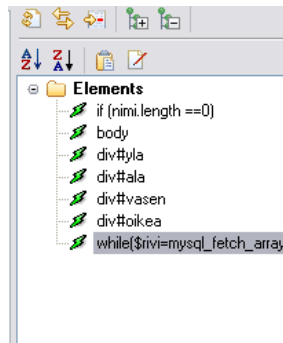
PSPad tukee myös sessioita: File-valikosta otetaan Sessions ja sen alta Save session. Tämä luo Sessions-valikkoon pikalinkin, jonka kautta aukeavat sessioihin tallennetut tiedostot. Sessionin sisältöön ei pysty muokkaamaan, mutta Save session-valinnan alapuolelta löytyvältä Session Manager-paneelistä pystyy poistamaan tai aukaisemaan sessioita. Huomion arvoista on, etteivät sessiot PSPadissa muodosta erillistä tiedostoa kuten projektit.

Vasemman reunan paneelistä löytyy välilehti, nimeltä FTP, kuten kuva 23 osoittaa. Painalla ensimmäistä näppäintä, joka on ainoana aktiivisena, saa näkyviin, kuvan 22 mukaisen, yhteysvalikon. New Connection-painikkeesta saa esiin kuvan 21 ikkunan. Testataksemme yhteyden käyttäytymistä, luomme yhteys1 nimisen yhteyden. Ensimmäisen napin vieressä olevasta nuolesta saamme listan yhteyksistä, jota klikkaamalla, ohjelma yrittää muodostaa automaattisesti FTP-yhteyden. Saman toiminnon saa myös Kuvan 22 valikosta valitsemalla yhteyden ja painamalla Connect. Kuvasta 24 näkee, että onnistunut yhteys aktivoi loput painikkeet valikosta. Näistä tärkeimpänä mainittakoon toinen vasemmalta, joka katkaisee yhteyden. Loput ovat normaaleja kansion muokkaustyökaluja. Yhteyden ollessa päällä, pystyy tiedostoja hakemaan, muokkaamaan ja tallentamaan.

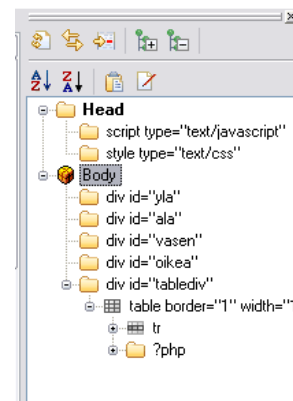
Kuvan 25 keskimmäisestä, vihreästä kansiorakennetta kuvaavasta, näppäimestä saa näkyviin oikeaan reunaan paneelin, joka esittelee tiedoston rakennetta. Kuvasta 27, vasemmalta laskien, ensimmäistä nappia painamalla pystyy päivittämään listan. Riippuen missä kohtaa kursori on tekstissä, päivittyy lista vastaamaan elementtien attribuutteja. Näiden listojen sisältö saattaa muuttua kuten kuvasta 26 näkee. Samassa kuvassa kursori on tyylien määrittämissä.



KUVA 25. Oikeanpuoleisen paneelin aukaisunappi.

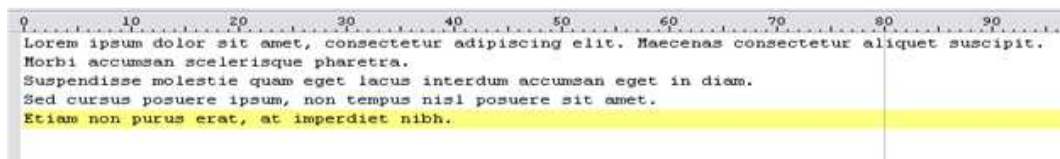


KUVA 26. Sivun elementtinäkymä.



KUVA 27. Sivun rakennennäkymä.

HTML-valikosta ottaessa Conversion ja Text to HTML antaa ohjelma automaattisesti tekstiin html-tagit. Ohjelma luo tällä uudelle tekstille uuden välilehden, jonka muoto on html, välittämättä mikä alkuperäinen muoto oli. Kuten kuvista 28 ja 29 voi päätellä, tämä keino on hyvä valmiin tekstin käsittelemiseen. Settings-valikon Program Settings-napin takaa olevasta alavalikosta voi päättää muuttaako html-muunnosrivinvaihdot rivinvaihdoksi
 vai kappaleiksi <p></p> seuraavasti. Tools-valikon ensimmäinen kohta ”HTML – text to HTML- Lines as Paragraphs ”. Raksi ruudussa tuottaa <p></p> tagit ja ruutu tyhjänä
 tagin.



KUVA 28. Lorem Ipsum-tekstiä



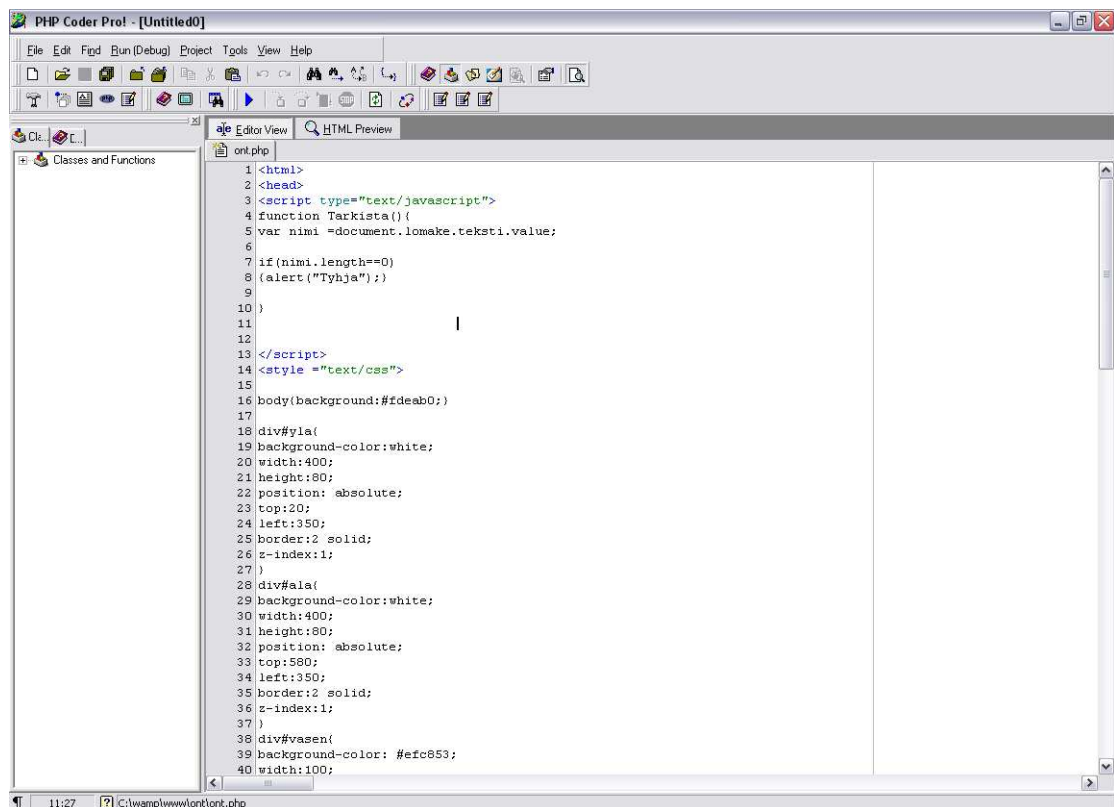
KUVA 29. Kuvan 28 Lorem Ipsun muutettuna automaattisesti html:läksi.

3.4 PHP Coder

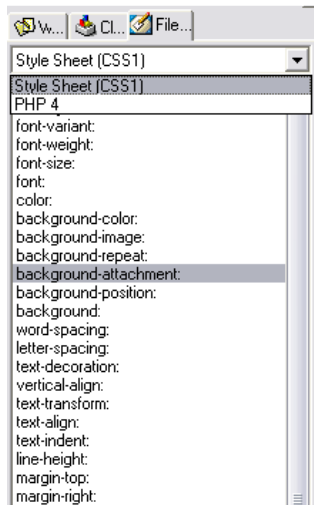
PHP Coder on nimensä mukaisesti erityisesti php-kielille kehitetty tekstieditori. Kuten kuva 30 osoittaa, sisältää se kaikki perusominaisuudet mitä tekstieditorilta odottaa, valitettavasti ohjelma ei tarjoakaan juuri muuta.

Rivinumeroitusta saa näkyviin View-valikosta Preferences and Settings-napin takaa. Editor-kohdasta ruksita Show Line Numbers-kohta.

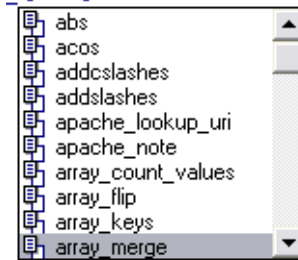
Ohjelma ei tarjoa minkäänlaista tukea JavaScriptille. Päätukiryhmiä ovatkin vain css1 ja php4. Huomautettakoon, että kirjoitushetkellä uusimmat versiot näistä ryhmistä ovat css2 ja php5. Yllä mainituille ryhmille löytyy ohjelman vasemmanpuoleisesta paneelista pikavalikko. Lista on kuitenkin turha, sillä varsinkin php-työskentelyssä on nopeampaa kirjoittaa lyhyet sanat käsin kuin valita ne listasta. Tyylielle ohjelma ei anna värikoodausta ollenkaan. Kuva 31 esittelee tyyliohjelmanäkymää.



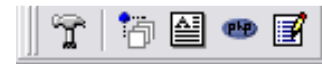
KUVA 30. PHP Coder-ohjelman yleiskuva



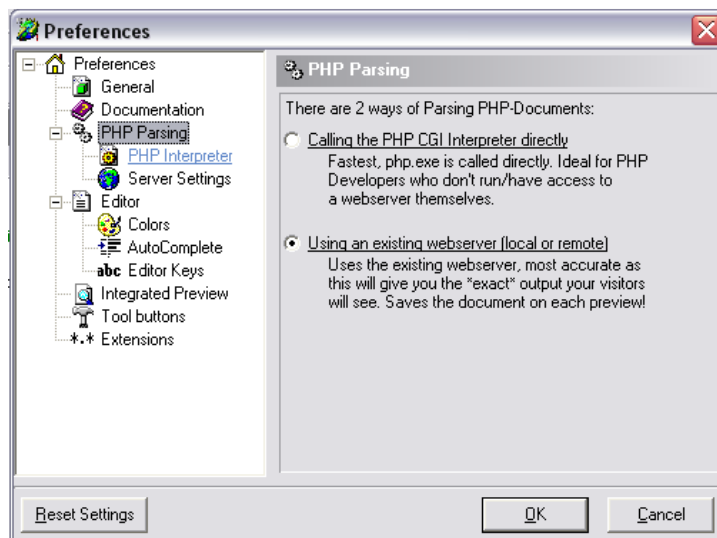
**KUVA 31. Attribuut-
tivalikko vasemman-
puoleisessa paneelis-**



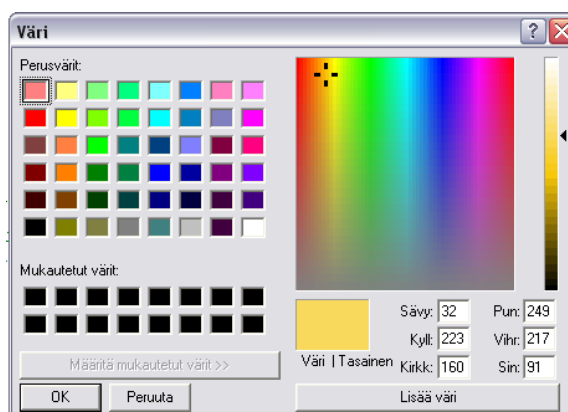
**KUVA 32. Koodikes-
keinen attribuuttiva-
likko.**



**KUVA 35. PHP Co-
der:in erikoisnäp-
päimiä**



KUVA 34. Php-parsimistavan valinta



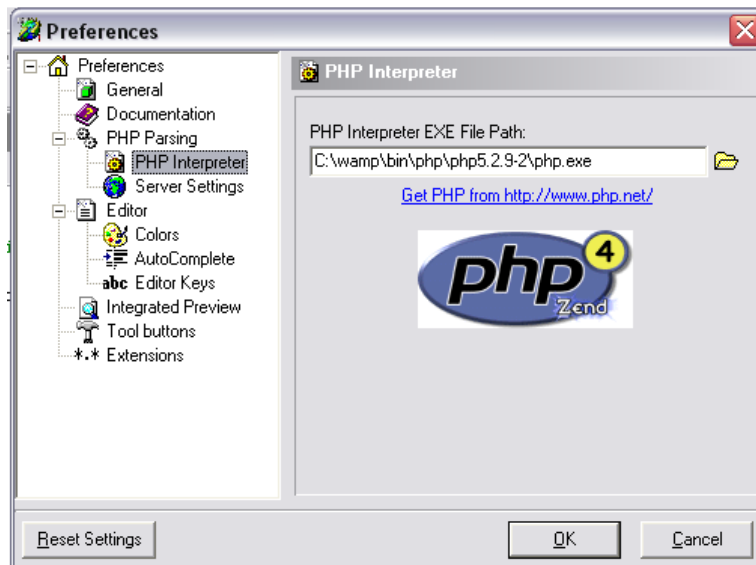
KUVA 33. Värivalikko

Automaattisen täydennyksen listan saa näkyviin painamalla yhtä aikaa Ctrl- ja väilyöntinäppäintä. Täydennys ei anna esille mitään sanan lisäksi, ja lista on muutenkin vajaa. Lista ei myöskään elä, vaan jumiutuu paikalleen siinä, missä kilpailevien ohjelmien lista elää kirjoitetun tekstin mukaan. Toisin sanoen listalta joutuu hakemaan sanoja käsin, joka auttaa kyllä oikeinkirjoituksessa, mutta vei paljon aikaa. Sanalistaa pystyy itse muokkaamaan Preferences and Settings-valikosta Editor-kohdan Auto-Complete-lehdeltä.

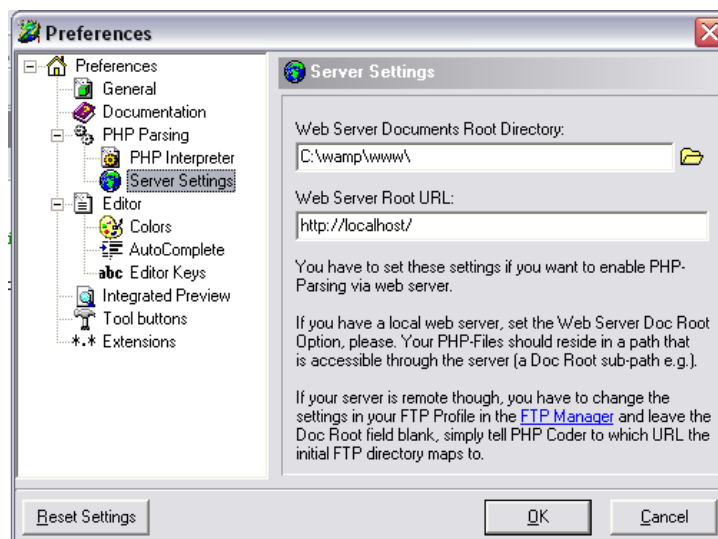
Yksi PHP Coder:in erikoisuuksista on muokattavat painikkeet. Ohjelma sisältää valmiiksi pienen määrän perus html-tageja painikkeiden takana ja oman värimikserin. Jos painikkeet eivät näy, raksita ruutu View-valikon Toolbars-alavalikon kohdasta HTML toolbars.

ToolButton Categories tuo esille kuvan 35 valikon. Valikon ensimmäinen työkalu vie Preference-valikkoon, josta muiden nappien sisältöä pystyy muuttamaan. Toinen nappi sisältää valmiita kaavake-elementtejä ja kolmas, html-elementtejä. Neljäs ja viides sisältävät kolme muokattavaksi tarkoitettua toimintoa, näitä voi halutessaan tehdä myös lisää. Neljäs on keskittynyt php-komentoihin ja viides erillisten ohjelmien käynnistykseen.

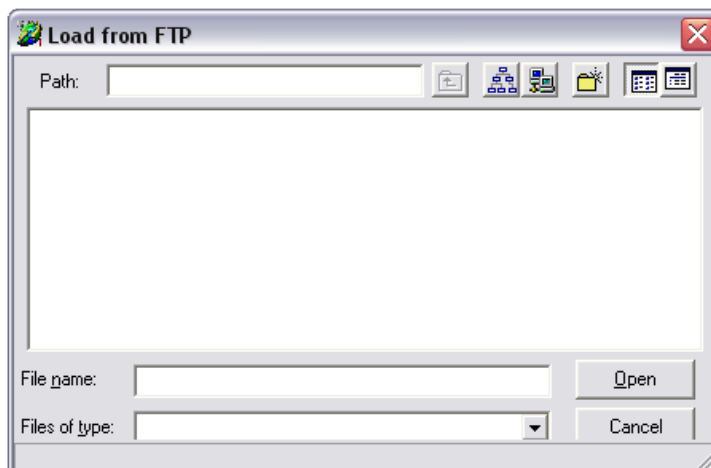
Ohjelma sisältää oman nettiselaimen, jolla koodia voi testata. Html toimii valmiina, mutta php ei. Preferences-valikosta löytyy kohta PHP Parsing, joka tarjoaa 2 vaihtoehtoa php-dokumentin parsimiseen. Koska käytössä on Wamp-serveri, käytetään tässä demonstraatioissa sitä eli kuvan 34 mukaisesti otetaan alempi vaihtoehto. PHP Interpreter-valikosta määritetään ohjelmalle php-tulkki, joka tässä tapauksessa sijaitsee Wamp-serverin alaisuudessa. Kuvasta 36 näkee, että käytössä on php:n versio 5.2.9.2. Server Settings-välilehdellä määritetään ensin tiedostojen juuri ja seuraavassa serverin juuren osoite. Tässä tapauksessa, kuten kuva 37 osoittaa, serverin juurikansio sijaitsee osoitteessa C:\wamp\www\ ja serverin juuri osoitteessa http://localhost/.



KUVA 36. Php-tulkin valinta



KUVA 37. Serverin asetusten määrittäminen.



KUVA 38. Yhteyden muodostusvalikko

FTP Accounts Manager

Profile name:

Login Information:

Host name: Port:

User name: Password:

Anonymous

Settings:

Initial Remote Dir:

Passive Mode

Server Settings

If you want your PHP files parsed on a remote server, you have to specify which remote directory maps to the server document root:

Doc Root Dir:

Root URL:

Proxy/Firewall Settings

Host name: Port:

KUVA 39. Yhteyden määrittämisvalikko



KUVA 40. Vasemman paneelin välilehtien avausvalikko.



KUVA 41. Projektivälilehti.

Jos nämä asetukset ovat oikein, pystyy ohjelma tulkitsemaan php-koodia itsestään HTML Preview-välilehdessä, edellyttäen, että tiedosto on tallennettu php muodossa ja on serverin alaisuudessa.

File-valikosta löytyvät Load From FTP ja Save to FTP ovat molemmat rakenteeltaan samanlaisia. Kuvan 38 toisesta, kahta tietokonetta esittävästä painikkeesta pääsee kuvan 39 valikkoon määrittämään FTP-asetuksia. Painamalla New FTP Site-nappia ohjelma jumittuu, oli kentissä tietoja tai ei. Tästä johtuen täysin varmaa toimivuutta ohjelman FTP-kyvyistä ei saada.

Jos vasemmanpuoleisessa paneelissa ei näy projektiosuutta, saa sen esiin painamalla kuvan 40 keskimmäistä nappia. Kuvan 41 viimeisestä aktiivisesta painikkeesta saa näkyviin tiedostoselain-välilehden. Erona muiden ohjelmien projekteihin, PHP Coder ei tarjoa mahdollisuutta siirtää auki olevia tiedostoja projektiin, vaan kaikki tiedostot lisätään Avaa-ikkunan kautta, joko Project-valikon tai hiiren oikealla saatavan valikon kautta.

3.5 Emerald Editor

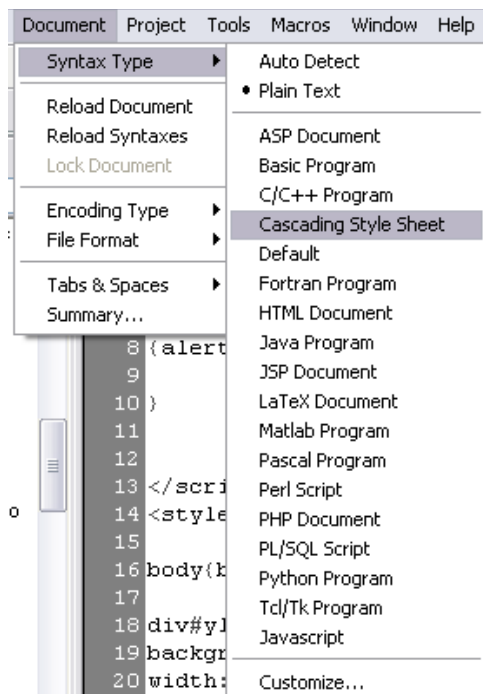
Kehutun Crimson Editor:in kehityksen loputtua vuonna 2004, siirtyi kehitys, avoimen lähdekoodin ansiosta, editorin yhteisön käsiin. Emerald Editor on Crimsonista vahvasti vaikutteita saanut versio, jonka kehitys jatkuu yhä. Vaikka kuvassa 42 ohjelman nimi näyttäisi olevan Crimson Editor, on kyse kuitenkin Emerald Editorista. (Crimson Editor, 2010).


```

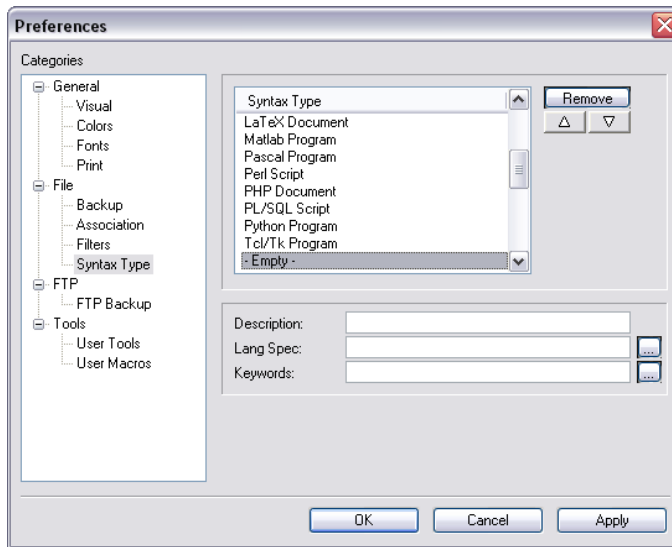
1 <html>
2 <head>
3 <script type="text/javascript">
4 function Tarkista(){
5 var nimi =document.lomake.teksti.value;
6
7 if(nimi.length==0)
8 {alert("Tyhjä!");}
9
10 }
11
12
13 </script>
14 <style ="text/css">
15
16 body{background:#fdeab0;}
17
18 div#yla{
19 background-color:white;
20 width:400;
21 height:80;
22 position: absolute;
23 top:20;
24 left:350;
25 border:2 solid;
26 z-index:1;
27 }
28 div#ala{
29 background-color:white;
30 width:400;
31 height:80;
32 position: absolute;
33 top:580;
34 left:350;
35 border:2 solid;
36 z-index:1;
37 }
38 div#vasen{
39 background-color: #efc853;
40 width:100;
41 height:580;
42 position: absolute;
43 top:45;
44 left:320;

```

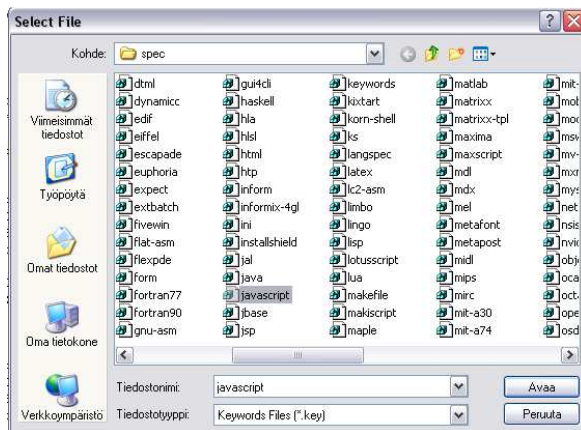
KUVA 42. Emerald Editor-ohjelman yleiskuva.



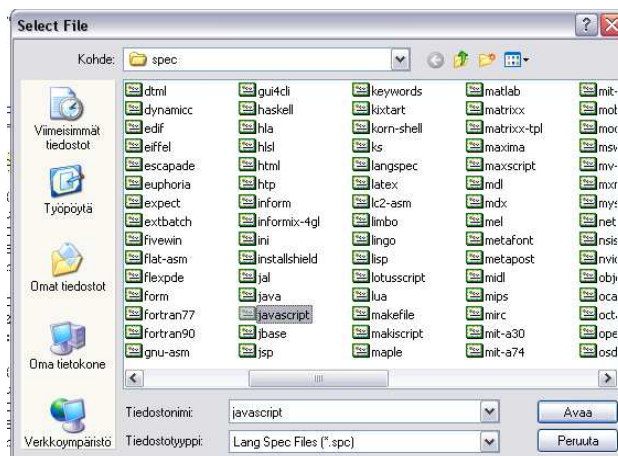
KUVA 43. Värisyntaksin vaihto.



KUVA 44. Värisyntaksin vaihtovalikosta valitaan tyhjä kieli, jota aletaan muo-



KUVA 45. Värisyntaksin kielen valinta.



KUVA 46. Värisyntaksin sanojen valinta.

Emerald Editor on valittu vertailuryhmään, koska se toimi ohjelmoinnin opetustyökaluna ennen kuin se korvattiin Notepad++:lla.

Värisyntaksin saa vaihdettua Document-valikon, Syntax Type-kohdan alavalikosta. Jos listassa ei ole tarvittavaa kieltä, on mahdollista, että kieli on kuitenkin saatavilla.

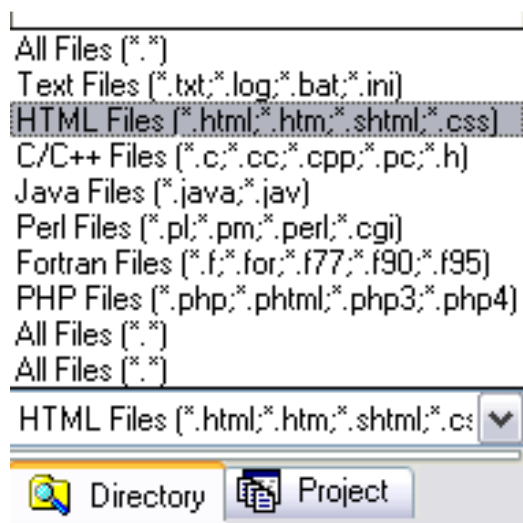
Kuvassa 43 listan viimeisenä on itse lisätty JavaScript-kielen värisyntaksi. Php-kielen kohdalla ohjelma ei osaa löytää värejä, vaan näyttää kaiken tekstin mustana, tämä johtuu vääristä määrittämissä, joita voi myös muuttaa alla olevan oppaan avulla.

Tools-valikon kautta pääsee Preferences-valikkoon, jonka Syntax Type-osiosta voidaan muokata ja lisätä värisyntakseja. Description eli kuvauskenttään tulee kielen nimi, tässä tapauksessa JavaScript. Lang Spec ja Keywords kenttiin tulee etsiä halutun kielen määrittäykset. Katso kuvat 45 ja 46.

Parien löytäminen toimii vain suluissa ja silloinkin sulut merkkautuvat pienellä alleviivaus-merkillä. Ohjelmassa koodin kirjoitus on hidasta, sillä ohjelma ei sisällä minikäänlaista automaattista täydennysmahdollisuutta.

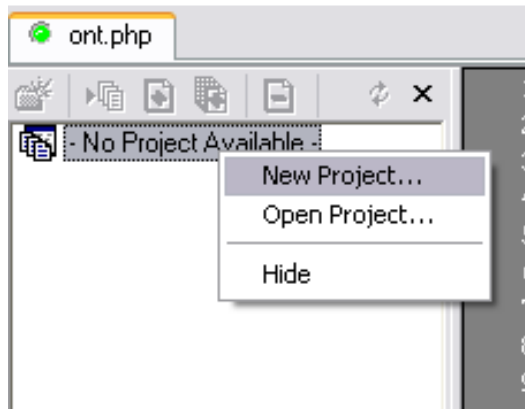


KUVA 47. Valikko.

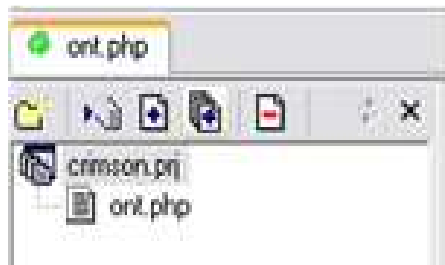


KUVA 48. Tiedostotyyppien suodatusvalikko.

Jos projektivalikko ei ole näkyvässä, saa sen näkyviin painamalla kuvan 47 keskimäistä näppäintä, jonka kuvauksessa lukee Project Window. Tällöin ohjelma avaa vasemman sivupaneelin, jossa toimivat sekä projektiominaisuudet, että tiedostoselain. Näiden ominaisuuksien välillä pystyy vaihtelevaan painamalla välilehtiä vasemmassa alakulmassa, katso kuva 48. Tiedostoja selatessa on mahdollista rajat näytettävät tiedostotyytit.

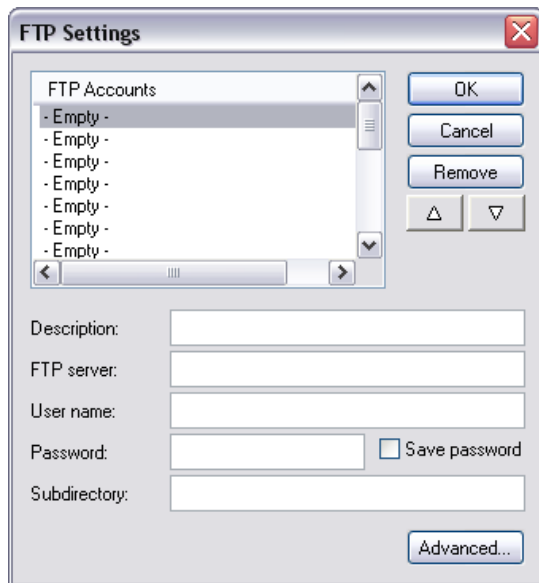


KUVA 49. Uuden projektin luonti.



KUVA 50. Uuden projektin luonti

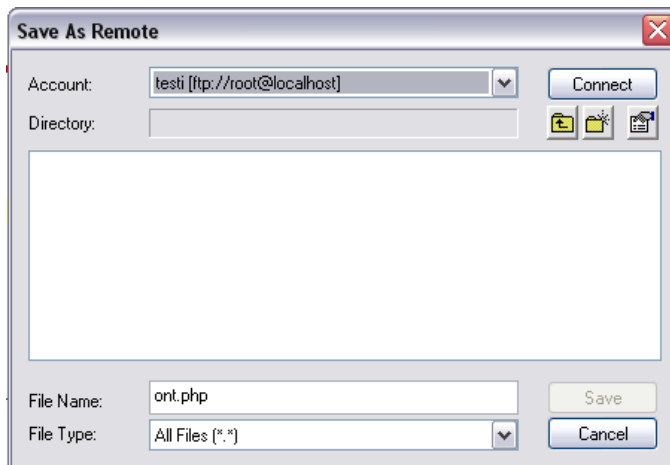
Uuden projektin voi luoda, kuvan 49 mukaisesti, painamalla – No Project Available — kohtaa hiiren oikealla näppäimellä ja valitsemalla New Project. Samasta valikosta voi myös avata jo olemassa olevan projektin. Seuraavaksi aukeaa ruutu, jossa määritetään projektin nimi ja asennuspaikka. Jos jo olemassa olevaa projektia yrittää avata ”normaalisti” File-valikon Open-toiminnolla, aukeaa projekti silloin tekstinä.



KUVA 51. Uuden FTP yhteyden luonti

Projektivalikon näppäimet ovat nähtävissä aktiivisena kuvassa 50. Ensimmäinen näppäin luo projektiin uuden kansion. Toisella näppäimellä aukeaa tiedostoselainikkuna, jonka kautta pystyy lisäämään projektiin haluttuja tiedostoja. Kolmas nappi lisää aktiivisen tiedoston projektiin ja neljäs kaikki tiedostot. Viides taas poistaa valitut tiedostot projektista, kuitenkin sulkematta niitä. Lisätessä tiedoston projektiin, tallentuu se projektin siihen osaa, joka on aktiivinen. Esimerkiksi, jos projektissa oleva kansio on aktiivinen, tallentuvat tiedostot sinne. Kansiokohtaisen projektivalikon saa klikkaamalla kansiota hiiren oikealla näppäimellä.

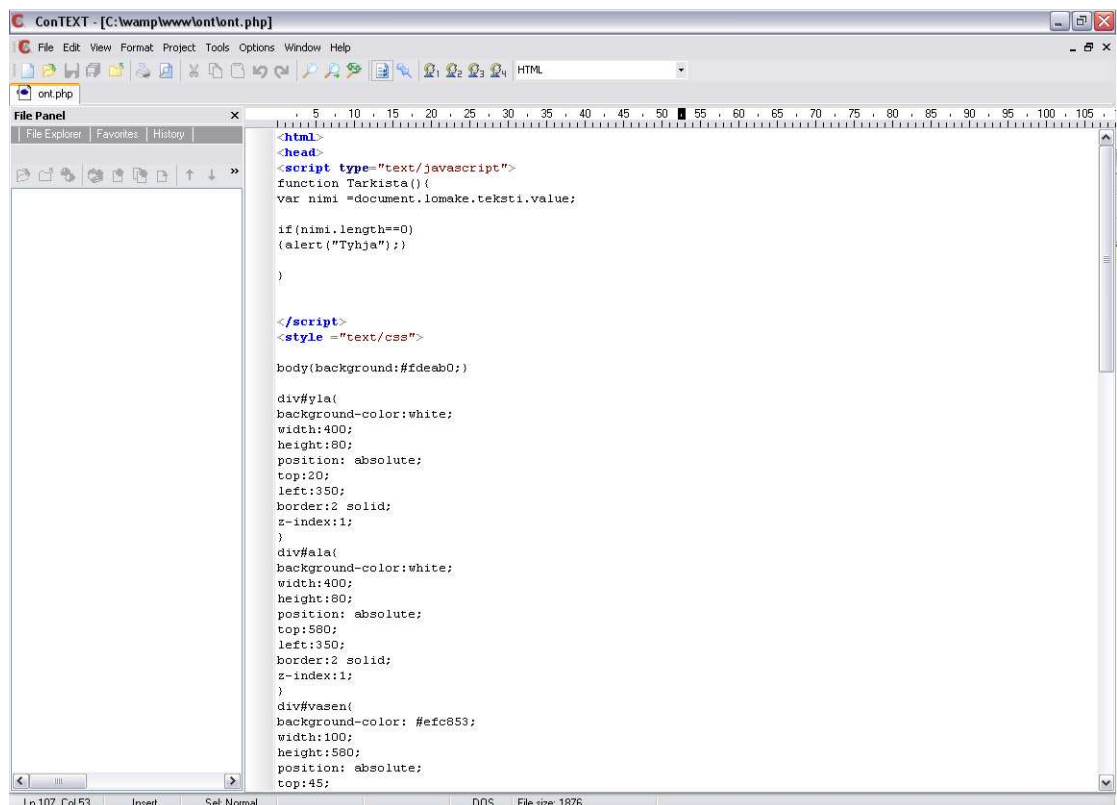
File-valikon FTP-alivalikosta valitessa FTP settings päästään määrittelemään yhteyttä. Esille tulee kuvan 51 valikko. Description-kohdan tekstistä tulee yhteyden nimi. Kuten kuvasta 52 näkee, File valikon FTP-osuudesta löytyvä Open Remote ja Save As Remote eivät eroa paljoa normaaleista avaa ja tallenna ikkunoista.



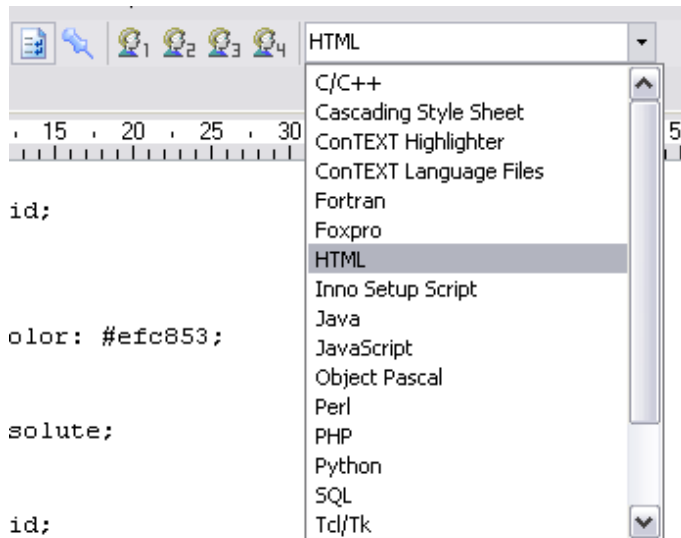
KUVA 52. FTP-yhteyden avaaminen.

3.6 ConTEXT

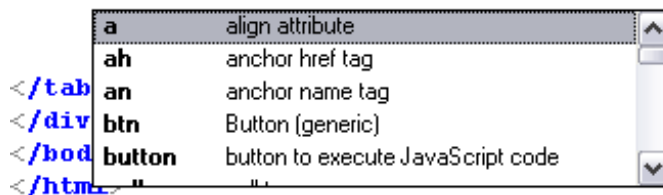
ConTEXT on avoimen lähdekoodin tekstieditori, joka täyttää ohjelmille asetettavat perusvaatimukset, mutta on sisällöllisesti hyvin vajaa.



KUVA 53. ConTEXT -ohjelman yleiskuva



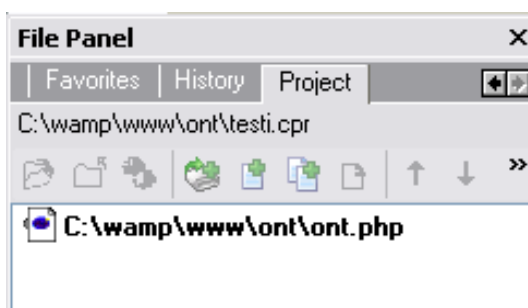
KUVA 54. Kielen vaihto



KUVA 55. Automaattinen tekstinsyöttö.

```
<form name="frm" method="GET" action="mailto:" onSubmit="JavaScript:return validate(this);">
  <input type="text" name="txt" value="">
  <input type="radio" name="rad" value="0">
</form>
```

KUVA 56. Esimerkki ConTEXT-ohjelman tekemästä kaavakkeesta



KUVA 57. Projektien hallinta.

Options-valikon Environment Options-alivalikon ensimmäiseltä välilehdeltä voi vaihtaa ohjelman kielen. Editor-välilehdeltä saa rivinumerot näkyviin raksittamalla kohdan Line numbers.

Valikkorivin lopusta löytyy tekstin väri- ja täydennyssyntaksi, josta voi valita halutun kielen. Kuva 54 demonstroi tätä. Vaikka ohjelma listaa useita kieliä, suurin osa niistä ei sisällä mitään muuta kuin värikoodauksen.

Ctrl- ja J-näppäimen yhdistelmällä saa näkyviin Kuvan 55 mukaisen valikon. Tämä esimerkki käyttää html-kieltä, koska se on ohjelmassa hyvin tuettu. Valikon attribuutit eivät ole sanoja vaan valmiita koodipohjia. Esimerkiksi valitsemalla form saa kuvan 56 mukaisen koodipohjan.

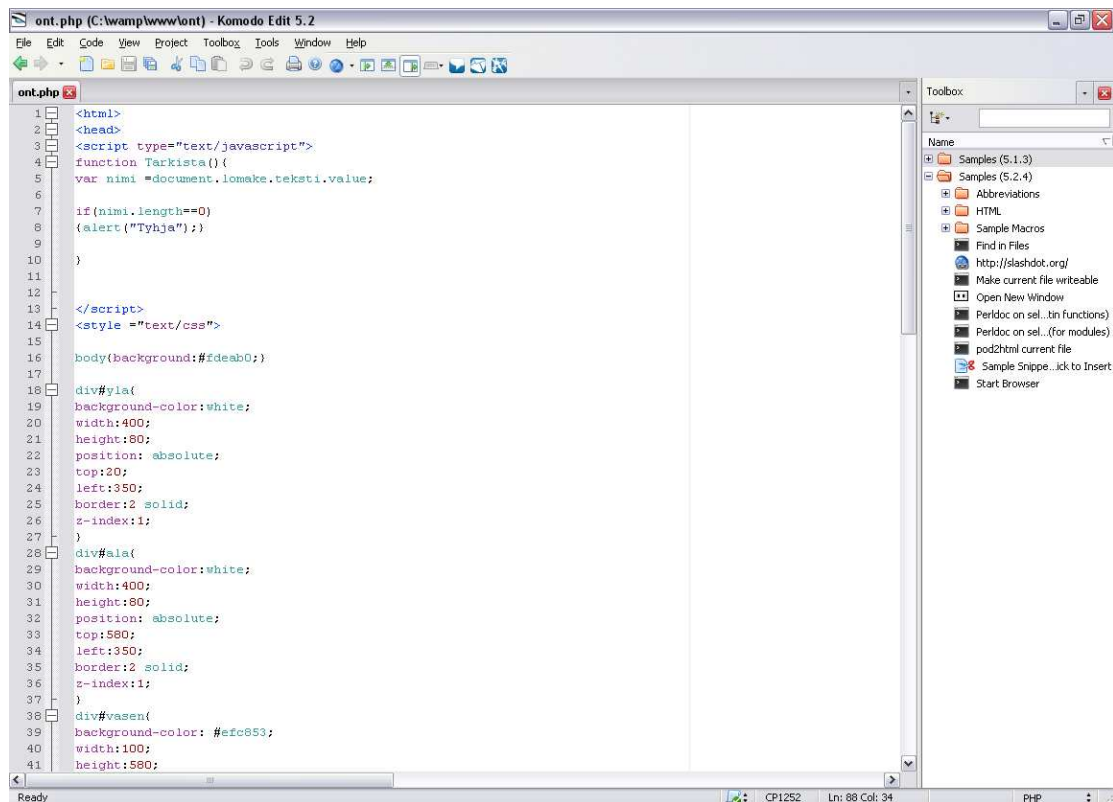
View-valikosta painamalla File panel-näppäintä saa esille vasemmanpuoleisen paneelin, josta löytyvät tiedostoselain, suosikit ja historiikki sekä, jos projektitiedosto on auki, sille oma välilehtensä. Kuten kuvasta 57 voi nähdä, projekti tiedosto ei tue kansiorakenteita, eikä ole muutenkaan kovin monipuolinen.

3.7 Komodo Edit

Komodo Edit on Komodo IDE-ohjelman kevyt versio. Tästä huolimatta tämä 33 megabitin ohjelma sisältää paljon ominaisuuksia ja on varteenotettava kilpailija markkinoilla. (ActiveState Software).

Ensimmäisellä asennuskerralla ohjelma ei pysynyt toiminnassa vaan kaatuili jatkuvasti. Poistettuani ohjelman ja asennettuani tämän uudestaan pysyi se vakaana. Kerran ohjelma ilmoitti testauksen aikana kaatumisesta sulkeutumisen yhteydessä.

Edit-valikosta valittaessa Preferences-alivalikko, löydetään Code Intelligence-välilehti. Välilehdeltä ruksita kohdat “Enable automatic autocomplete and calltip triggering while you type” ja “Enable autocomplete fill-up characters”. Näistä ensimmäinen mahdollistaa sanojen ennustamisen ja toinen merkkien, kuten lainausmerkkien, täydentämisen. Editor-välilehdeltä otetaan käyttöön ”Show line numbers”. Servers-välilehdeltä pystyt määrittämään FTP-yhteyden asetukset.



KUVA 58. Komodo Edit-ohjelman yleiskuva

Suljettavaa tagia kirjoitettaessa, osaa Komodo suositella tagiparia tälle kuvan 59 mukaisesti. Laukaisija on /-merkki. Automaattisen täydennyksen laatu on kaikista testatuista ohjelmista paras. Komodo tunnistaa kirjoitettavan kielen ja ehdottaa täydennyksiä sen mukaan ilman, että käyttäjän tarvitsee itse vaihtaa kieltä valikoiden kautta.

Täydennyslistatkin ovat kattavia. Kuvan 60 esimerkissä lista tulee esille painamalla välilyöntiä ja näyttää border-attribuuttien asetukset. Kun käytössä on sulkujen auto-



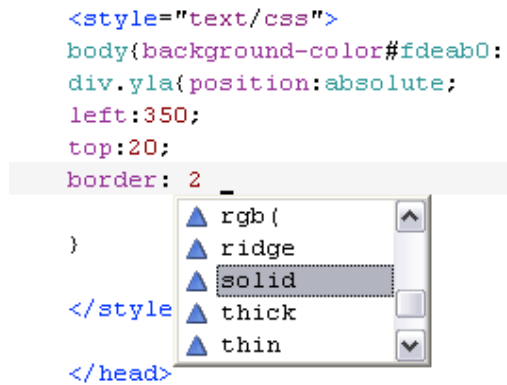
KUVA 59. Lopettavan tagin täyttö.

maattinen paritus, antaa ohjelma komennon kuvauksen ja sulkujen täyttöohjeen. Kaarisulkujen paritus ei kuitenkaan toimi kirjoitettaessa sivun tyyliohjaa.

```

<style="text/css">
body{background-color#fdeab0:
div.yla{position:absolute;
left:350;
top:20;
border: 2
}
</style>
</head>

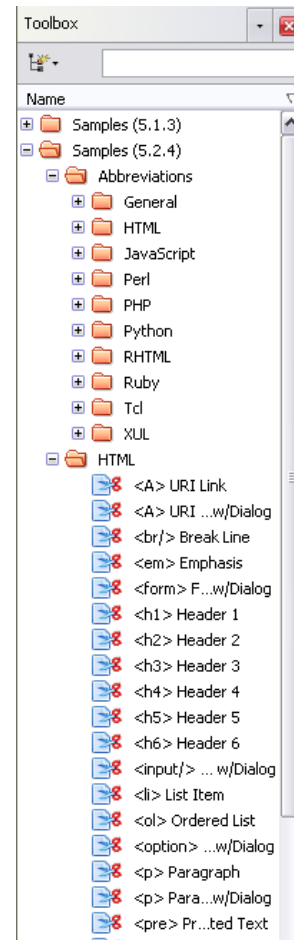
```



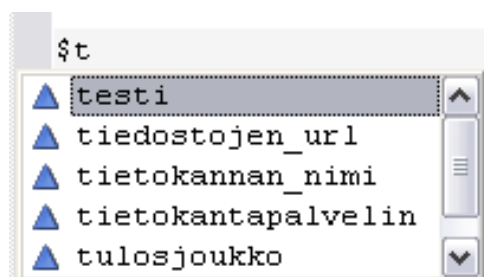
KUVA 60. Attribuutin täydennys.



KUVA 62. Projektivalikko.



KUVA 61. Oikeanpuoleinen paneeli.



KUVA 63. Testi-niminen muuttuja löytyy ennustuksesta vaikka sitä ei ole itse dokumentissa.

View-valikon View as Language-alavalikosta voidaan valita haluttu kieli, mutta testi-käytössä tähän valikkoon ei ollut tarvetta koskea, sillä ohjelma väritti eri kielet ja osasi ennustaa tarpeellista kieltä.

Php-muuttujien kohdalla osaa Komodo muistaa avattujen tiedostojen muuttujat ja ne ilmestyvät tekstin ennustukseen muiden muuttujien rinnalle. Kuvassa 63 näkyy muuttuja \$testi, vaikka sitä ei ole varsinaisessa dokumentissa, vaan tallennetussa testi.php-tiedostossa, joka on avattu Komodolla. Ohjelma osaa myös päivittää muuttujan jos, Komodo Edit:iä käyttäen, sen nimi alkuperäisessä tiedostossa muutetaan tai poistetaan.

Komodo osaa automaattisesti sisentää kirjoitettavat tagit. Ominaisuus ei kuitenkaan toimi suoraan liitettävään tekstiin. Ominaisuudesta saa eniten irti, kun aloittaa sivuston tekemisen kokonaan Komodo Edit:llä.

Perinteistä tiedostoselainta ei Komodo Editistä lyödy, kuitenkin View-valikon Toolbars-alavalikosta voi valita esille Open/find- paneelin. Tämä paneeli sisältää kaksi päätoimintoa. Se osaa muodostaa pudotusvalikon kaikista valitun kansion avattavista tiedostoista ja etsiä asetetusta kansioista hakusanalla hakusanan ilmentymiä ja esittää ne rivikohdittain.

Komodon projektitiedostot eivät varsinaisesti ole tiedostojen kokonaisuuksia vaan luodessa projektin Komodo liittyy siihen kaikki tiedostot, jotka ovat samassa kansiossa kuin projektitiedosto sekä näiden alikansiot tiedostoineen. Projektien luonti ja hallinnointi sujuu ylhäällä olevan Project-valikon kautta. Kuvan 62 oikean puoleisesta napista saa esille tiedostojen lisäsvaihtoehdot.

File-valikon Save Remotely As...- ja File-valikon Open...-alavalikon Remote file-kohdat, antavat molemmat samankaltaisen ikkunan. Ohjelma ottaa yhteyden heti kun serverilistasta on valittu haluttu yhteys. Listan vieressä oleva nappi vie Preferences-valikon Server-välilehdelle, josta pystyy säätämään yhteyksien asetuksia. Komodon ja Emerald Editorin FTP-ominaisuudet ovat lähes identtiset.

Ohjelman asetuksissa on mahdollisuus antaa sille serverin php-tiedostot, joita se voi käyttää php:n esikatseluun. Tästä huolimatta ei ohjelma osaa tulkita koodia vaan ohjelma näyttää vain koodin html-osuudet. Esikatseluominaisuuden saa päälle View-valikosta valitsemalla Preview in browser.

Kuvan 64 ensimmäiset kolme näppäintä aukaisevat paneelin soittamaansa suuntaan. Kolmannesta aukeaa oikeaan reunaan, kuvan 61 mukainen, esimerkkipaneeli. Paneelista voi valita valmiita elementtejä tuplaklikkaamalla niitä. Elementti ilmestyy kohtaan, jossa osoitin koodissa on. Nämä esimerkit ovat kattavia ja sisältävät kaiken mitä funktio tarvitsee, esimerkiksi taulukon, johon on valmiiksi rakennettu yhdelle riville yksi solu ja kaikki tarvittavat lopettavat tagit.



KUVA 64. Paneelivalikko.

4 OHJELMIEN VERTAILU

Useilla ohjelmilla on nähtävissä yhteisiä piirteitä, tämä koskee kaikkia ehdokas ohjelmia ja osaa karsiutuneista. Kaikki ehdokasohjelmat on rakennettu tyypillisen Windows-ohjelman muotoon, jossa ylimpänä on tekstipohjaisia pudotusvalikkoja ja sen alapuolella toimintokuvakkeita. Näiden alapuolelta löytyy välilehtirivi.

Packardin ja Racen (2003, 62 - 63) mukaan useat ohjelmien ominaisuudet ovat samoja joka ohjelmassa. Ero on toteutuksessa, joka ei välttämättä edes ilmene ohjelman käyttäjälle. He painottavat myös, että toiminnot ovat opetuksessa tärkeämpiä kuin niiden toteutus. Ohjelmien ainoat sitovuudet koodiin ovat pääasiassa ohjelmakohtaiset projektitiedostot, jotka voivat perustua erilaiseen rakenteeseen vaikka niiden toteutustekniikka, usein xml, olisivatkin samoja. Ulkonäöllisesti ohjelmat muistuttavat toisiaan, kuten kuvista 2, 7, 12, 30, 42, 53 ja 58 voi todeta.

Ohjelmien ominaisuudet löytyvät usein paneeleista. Käytetyin näistä on usein sijoitettu vakiona ohjelman vasempaan reunaan ja ns. lisäpaneeli oikeaan reunaan. Paneeli sisältää useimmiten useita ominaisuuksia jaettuna sen sisäisille välilehdille. Emerald Editor, ConTEXT ja Komodo Edit ovat ainoita ohjelmia, joissa paneeleita ja niiden sisäisiä välilehtiä ei voi siirtää. Muissa ohjelmissa paneelin paikkaa voi halutessaan vaihtaa mihin ohjelman reunaan tahansa tai tehdä niistä leijuvia. Tämä mahdollistaa ohjelman personoimisen, mikä ei välttämättä opetuksen kannalta ole hyvä ratkaisu, sillä opettajan ja oppilaan toiminnot voivat olla eri paikoissa, mutta itsenäiseen työhön toiminta antaa helpotusta. Programmer's Notepad on vienyt paneelien käytön niin pitkälle, että antaa paneelien muodostua sisäkkäin yhdeksi paneeliksi, jossa ominaisuudet ovat välilehdillä.

Kuten jo alussa todettiin, tekniikat ovat tällä hetkellä melko vakiintuneita sivuston tekemisessä. Käytössä on perusrakennuskieli html, sen tyyliasetuksia säättävä css ja komento- ja sovelluskielenä php, joka osaa kommunikoida MySQL-tietokannan kanssa. Edellä mainittuja voidaankin pitää peruskriteereinä ohjelmaa valitessa. Näille ohjelman tulee tarjota mahdollisimman kattavat tuet. Ohjelmointitehtävissä melkein kaikki ohjelmat on optimoitu ymmärtämään vaadittavia kieliä. Poikkeuksena PHP Co-

der, joka ei ymmärrä JavaScriptiä millään tasolla ja PSPad, jonka JavaScript tuki ei toimi vaaditulla tavalla. PHP Coderista ei myöskään löydy värikoodausta tyylielle, vaikka niitä kohtalaisesti ymmärtääkin.

Suurimmat erot ohjelmissa syntyvät, kun aletaan vertailla kuinka paljon ohjelma osaa tuottaa ja käsitellä tekstiä käyttäjän puolesta. Tähän lukeutuvat automaattiset tekstin käsittelyt ja sisennykset sekä automaattinen tekstinsyöttö. Vertailtavista ohjelmista parhaiten automaattinen tekstinsyöttö on edustettuna PSPadissa. PSPadille uniikkeja ominaisuuksia ovat sulkujen automaattinen lisääminen funktioon, ominaisuus, jota näkee todella harvoin, jopa kaupallisissa ohjelmissa. PSPadin tapa antaa käyttäjän valita html-tageille ja tyyli pohjan ominaisuuksille attribuutteja listalta on huikea. Oppilas näkee kerralla mitä ominaisuuksia ko. elementti voi saada ja pystyy säätämään niitä helposti. Ominaisuus voi pahimmillaan sitoa oppilaan liaksi ohjelmaan, jolloin ohjelmointityö voi muuttua koodikeskeisyyden sijaan ohjelmakeskeiseksi. PSPadilla on myös ohjelmista paras tapa käsitellä sivuille menevää tekstiä, niin varsinaista tekstiä kuin Lorem Ipsum täytetekstiä, jota se pystyy itse tuottamaan.

Komodo Editin erottavaksi tekijäksi muodostuu kyky osata ilman kielenmääritelmän vaihtoa ehdottaa php-muuttujia, tyylien ominaisuuksia ja sulkevia html-tageja. Notepad++ ja Programmer's Notepad sisältävät myös hyvän automaattisen tekstinsyötön, kuitenkin ominaisuus tekee vain sen mitä nimi antaa ymmärtää, päättää käyttäjän kirjoittaman sanan.

Projekti ominaisuus löytyi myös suurimmasta osasta ohjelmia. Ainut, jossa sitä ei ollut, oli PHP Coder. Projekti ominaisuus tyyppiä ilmeni kahdenlaisia: sessiopohjaisia ja projektipohjaisia. Sessiot eroavat projekteista niiden luonteen takia. Sessioissa ohjelma tallentaa auki olevien tiedostojen nimet muistiin ja osaa avata ne kaikki kerralla, kun taas projekteille pystyy antamaan kansiorakenteita. Projekteille on myös ominaista, että niitä pystyy tallentamaan tiedostoksi, jonka pääte on ohjelmakohtainen. Projektitiedostot soveltuvat, kansiorakenteen tukemina, loistavasti portfolion kokoamiseen, jolloin opettaja saa kerralla näytille kaikki arvioimista kaipaavat tiedostot. Tämän ominaisuuden kohdalla mikään ohjelma ei tee selvää pesäeroa muihin.

Vain kolme ohjelmaa vertailussa tarjoaa mahdollisuuden katsella valmista sivua selainikkunassa. Nämä ohjelmat ovat PHP Coder, Komodo Edit ja PSPad. PHP Coder on ainut, joka osaa onnistuneesti kääntää php-kieltä, kunhan serverin ominaisuudet on määritetty oikein. Tämä ominaisuus on PHP Coderin parasta antia, ohjelman muilta osin jääden kilpailijoiden jalkoihin. Testauksen aikana ei Komodo Editorilla sama ominaisuus suostunut toimimaan ollenkaan, vaikka serverin asetukset olivatkin oikein. Tämän ja PSPadin html-koodin toisto tyyleineen ja JavaScripteineen ovat hyvin identtiset.

Edellisen luvun ohjelmien lisäksi Emerald Editorista löytyy FTP-toiminto. PHP Coder:ssa toimintoa ei pystytty testaamaan, joten sitä ei tähän vertailuun pystytä ottamaan. Komodo Edit ja Emerald Editor luottavat perinteiseen Avaa-ikkunaan siinä missä PSPad tuo FTP-yhteyden päässä olevat tiedostot ja kansiot paneeliin. Kaikissa ohjelmissa FTP-yhteyden asetukset on sijoitettu yhteyden luonnin läheisyyteen.

Tiedostopuu-ominaisuuden musta lammas on Komodo Edit. Siinä missä muut ohjelmat ovat kopioineet ominaisuuden resurssien hallinnasta, on Komodo Edit muuttanut sen luonnetta pudotusvalikoksi ja liittänyt kylkeen tiedostojen sisällöllisen sanahaun. Ominaisuus on hyödyllinen erityisesti osaavissa käsissä, mutta muuten se voi vaikuttaa jopa hankalalta käyttöä.

Suomen kielelle ohjelmista saa kolme, Notepad++:san, PSPadin ja ConTEXTin. Näistä mikään ei ole 100 % käännetty, vaan mukana on joko kääntötyön jälkeen tullutta tekstiä, joka tarkoittaa sitä, ettei käänнос ole ajantasainen tai suomennokset ovat kehoja. Nämä miinustekijät ovat kuitenkin pieniä vikoja kokonaiskuvassa, sillä suomen kieli toimii kaikissa ohjelmissa käytännössä hyvin ja selkeyttää ohjelmaa tarvittavilta osin.

Kuten rivien välistä, ja ohjelmien nimien mainintojen määrästä, voi päätellä, nousee kaksi ohjelmaa ylitse muiden. PSPad ja Komodo Edit. Näiden lisäksi Programmer's Notepad ja Notepad++ ovat hyvinkin varteenotettavia ehdokkaita valitessa opetusohjelmaa. PSPadin näyttämät eri elementtien ominaisuudet ja niiden helppo ja nopea säätö ovat varsin hyödyllisiä, tarkkaili asiaa miltä kantilta tahansa. Tämä poistaa käyttäjältä

tarpeen lähteä etsimään netistä ominaisuuksien nimiä ja asetuksia. PHP Coder on joukkion heikoin lenkki, eikä oikeastaan sovellu opettelu- eikä ammattikäyttöön. Taulukko 1:ssä on esitetty kaikkien vertailtavien ohjelmien vertailtavat ominaisuudet.

PSPadin heikkous onkin lähes olematon JavaScript-tuki, mutta mikään ohjelma ei loistanut ominaisuutta testatessa. Missään ohjelmassa ei ole ominaisuutta, joka auttaisi JavaScriptiä rakentamaan testikoodissa olevan nimi muuttujan rakenteen.

	Notepad++	Programmer's Notepad	PSPad	PHP Coder Pro	Emerald Editor	ConTEXT	Komodo Edit.
Htm1-yymmärrys	x	x	x	x	x	x	x
Php-yymmärrys	x	x	x	x	x	x	x
Css-yymmärrys	x	x	x	x	x	x	x
Javascript-yymmärrys	x	x	x	x	x	x	x
Väילהdet	x	x	x	x	x	x	x
Ominaisuudet paneelissa	x	x	x	x	x	x	x
Ennustava tekstinsyöttö	x	x	x				x
Sisennys toiminto			x				x
Projektimahdollisuus	x	x	x	x	x	x	x
Esikatselu			HTML	HTML/PHP			HTML
FTP			x		x		x
Tiedostosalain	x	x	x	x	x	x	x
Saatavana suomeksi	x		x			x	

TAULUKKO 1. Ohjelmien ominaisuuksien vertailu.

5 PÄÄTÄNTÖ

Opinnäytetyön tekeminen alkoi noin vuosi sitten. Arto Väätäinen, internetsovelluskehityksen opettaja, totesi, että silloiselle opetusohjelmalle Crimson Editorille pitäisi saada seuraaja ja että aiheesta saisi jopa opinnäytetyön. Jonkin aikaa mietittyäni kysäisin, että onko kukaan ottanut aihetta vielä. Ei ollut, joten se päättyi minulle. Opinnäytetyösuunnitelman laadittuani olin valmis aloittamaan, kunnes sainkin töitä. Tästä johtuen työ jäi roikkumaan ja sille oli vaikeaa löytää aikaa ja huonon pohjustuksen takia, itse tekeminenkin oli epäselvää. Suunnitelma, siitä mitä piti tehdä oli selkeä ja vertailtavat ohjelmistot valittu, mutta itse prosessin käyntiin saanti tuntui lähes ylivoimaiselta. Vuoden vaihteen jälkeen asiat alkoivat edistyä, kun työsopimus loppui ja opinnoista oli jäljellä enää vain opinnäytetyö. Nyt oli aikaa ja motivaatiota ottaa asioista kunnolla selvää ja ruveta tekemään. Kuitenkin aikaa kului jälleen suunnittelemattomuuteen ja yhdenmukaisen vertailun puuttumiseen, joten monia asioita jouduttiin jättämään pois. Opinnäytetyössä olisi esimerkiksi voinut verrata, Packardin ja Race (2003, 78) ehdotuksesta, eri ammattikorkeakoulujen ja IT-firmojen käytössä olevia ohjelmistoja ja ottaa ne mukaan vertailuun. Packard ja Race (2003, 78 - 80) kehottavat myös keräämään palautetta ja arviointeja ohjelmasta oppilailta. Tämä ei kuitenkaan ole juuri mahdollista opinnäytetyön toteuttamisen puitteissa, mutta sitä voidaan jälkikäteen soveltaa hyväksyttämällä ohjelma ryhmällä oppilaita.

Opinnäytetyön aihetta voi jatkaa esimerkiksi vertailemalla Javan kehitystyökaluja toisiinsa. Käytännössä tämä tarkoittaisi kahden suurimman ja suosituimman ohjelman, Netbeansin ja Eclipsen keskinäistä vertailua. Vertailua voisi tehdä esimerkiksi kuinka Javan ja Mobiili-Javan koodaustapa eroaa ohjelmien välillä ja mitkä ovat ohjelmien suurimmat käytännön erot. Muutaman vuoden sisään julkaistava html5-tekniikka voi muuttaa oleellisesti tässä opinnäytetyössä käsiteltyjen ohjelmien asemaa markkinoilla, silloin nähdään tosissaan, kuinka sopeutumiskykyisiä tekstieditorit ovat kehittyvän tekniikan alla. Siinä missä toiset kehittävät erilaisia käytettäviä tekniikoita, on ammattikorkeakoulun tietojenkäsittelijän tehtävä opetella käyttämään niitä, samaistua niihin ja kehittää tekniikoiden käytettävyyttä.

4 LÄHTEET

Kalli, Pekka 2003, Kotila, Hannu (toim). Ammattikorkeakoulupedagogiikka. Helsinki: Edita Prima Oy.

Packard, Nick (toim), Race, Phil (toim) 2003. Käytännön vinkkejä opetustyöhön . Hamina: YRITYSSANOMA OY.

Taube, Karin 1998. Portfolio. Jyväskylä: Gummerus kirjapaino OY.

Koivulahti-Ojala, Mervi 2001. ATK-opettajan opas. Jyväskylä: Gummerus kirjapaino OY.

Meisalo, Veijo, Sutinen, Erkki, Tarhio, Jorma 2003. Modernit oppimisympäristöt. Pieksämäki: RT-Print Oy.

Mikä on paras editori ohjelmointitarkoitukseen? 2010 Mureakuha.com. WWW-dokumentti. <http://wiki.mureakuha.com/wiki/>-

Mikäs_on_paras_editori_ohjelmointitarkoitukseen%3F. Päivitetty 8.2.2010. Luettu 21.4.2010.

Koodarin vaikea valinta kuusi hyvää editoria. 2003 Mikropc.net. WWW-dokumentti. <http://mikropc.net/nettilehti/pdf/2702200344.pdf>. Päivitetty 3.2003. Luettu 21.4.2010.

Tommi Myrskykari 2009. Kotisivut. WWW-dokumentti.

<http://myrskykari.tripod.com/index.html>. Päivitetty 27.08.2009. Luettu 21.4.2010.

Notepad++. Wikipedia.org. 2010. WWW-dokumentti.

<http://en.wikipedia.org/wiki/Notepad%2B%2B>. Päivitetty 13.4.2010. Luettu 21.4.2010.

Programmer's Notepad. Wikipedia.org. 2010. WWW-dokumentti.

http://en.wikipedia.org/wiki/Programmer's_Notepad. Päivitetty 12.12.2009. Luettu 21.4.2010.

PSPad. Wikipedia.org. 2010. WWW-dokumentti.

<http://en.wikipedia.org/wiki/Pspad>. Päivitetty 1.2.2010. Luettu 21.4.2010.

Crimson Editor. Wikipedia.org. 2010. WWW-dokumentti.

http://en.wikipedia.org/wiki/Crimson_Editor. Päivitetty 1.1.2010. Luettu 21.4.2010.

Komodo Edit: Free Open Source Dynamic Languages Editor. ActiveState Software 2010. WWW-dokumentti.

http://www.activestate.com/komodo_edit/. Ei päivitystietoja. Luettu 21.4.2010.

**LIITE 1(1).
Sivuston lähde-
koodi.**

```
<html>
<head>
<script type="text/javascript">
function Tarkista (){
var nimi =document.lomake.teksti.value;

if (nimi.length==0)
{alert ("Tyhja");}

}
</script>
<style ="text/css">

body{background:#fdeab0;}

div#yla{
background-color:white;

width:400;
height:80;
position: absolute;
top:20;
left:350;
border:2 solid;
z-index:1;
}
div#ala{
background-color:white;
width:400;
height:80;
```

**LIITE 1(2).
Sivuston lähde-
koodi.**

```
position: absolute;
top:580;
left:350;
border:2 solid;
z-index:1;
}
div#vasen{
background-color: #efc853;
width:100;
height:580;
position: absolute;
top:45;
left:270;
border:2 solid;
z-index:2;
}
div#oikea{
background-color: #efc853;
width:100;
height:580;
position: absolute;
top:45;
left:720;
border:2 solid;
z-index:0;
}
div#tablediv {
background-color: white;
position: absolute;
top: 120px;
```

LIITE 1(3).

Sivuston lähdekoodi.

```
left: 400px;
width: 290px;
height:170px;
border:2 solid;
z-index:0;
}
</style>
</head>
<body>
<div id="yla"></div>
<div id="ala"></div>
<div id="vasen"></div>
<div id="oikea"></div>
<div id="tablediv">
<table border="1" width="100%">
<tr ><td colspan="3">
<form action="ont.php" method="post" name="lomake">
<input type="text" name="teksti">
<input type="submit" onclick="Tarkista ()" value="Paina">
</form>
</td></tr>
<?php
$tietokantapalvelin = "localhost";
$skayttajatunnus = "root";
$salasana = "";
$tietokannan_nimi = "puhelinluettelo";
$yhteys = mysql_connect ($tietokantapalvelin, $skayttajatunnus, $salasana);
mysql_select_db ($tietokannan_nimi, $yhteys);
$sql_lause ="Select * from Luettelo WHERE `Nimi` LIKE '$_POST[teksti]!";
$stulosjoukko=mysql_query ($sql_lause);
while ($rivi=mysql_fetch_array ($stulosjoukko, MYSQL_ASSOC)) {
```

**LIITE 1(4).
Sivuston lähde-
koodi.**

```
print ("<tr><td>$rivi[ID]</td>  
<td>$rivi[Nimi]</td><td>$rivi[Puhelinnumero]</td></tr>");  
}  
?>  
</table>  
</div>  
</body>  
</html>
```