Minna Maltari

Uusi malli web-ohjelmoinnin opintojakson toteutuksessa

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Mediatekniikan koulutusohjelma

Insinöörityö

15.5.2013

Metropolia

Tekijä Otsikko Sivumäärä Aika	Minna Maltari Uusi malli web-ohjelmoinnin opintojakson toteutuksessa 46 sivua + 1 liitettä 15.5.2013
Tutkinto	insinööri (AMK)
Koulutusohjelma	mediatekniikka
Suuntautumisvaihtoehto	digitaalinen media
Ohjaaja	yliopettaja Kari Aaltonen

Insinöörityössä oli tavoitteena koostaa web-ohjelmoinnin opintojakso käyttäen uusia opetusmalleja ja näin parantaa ohjelmoinnin opetusta. Opintojakso toteutetaan käänteisen luokkahuoneen mallin mukaisesti, jossa opiskelija omaksuu opintojakson teoriaosuudet videotutoriaalimateriaalin ja harjoitustehtävien myötä. Lähiopetustunnit keskittyvät aktiiviseen opitun soveltamiseen ja kertaamiseen tehtävien ja kysymysten avulla.

Insinöörityössä etsittiin opintojakson opetussisältöön soveltuva videotutoriaalimateriaali korvaamaan aiemmin käytössä olleet teorialuennot. Teoriamateriaalin jakeluun ja yhteisölliseksi verkkotyökaluksi otettiin käyttöön oppimisalusta, jolle opetusmateriaalit ja tehtävät koostettiin. Lähiopetustyöskentelyä varten etsittiin www-pohjainen yhteisötyökalu.

Toteutettavan opintojakson laajuus on 14 opiskeluviikkoa, ja sen opetus käsittää ohjelmoinnin perusteet sekä oliopohjaisen ohjelmoinnin Python-ohjelmointikielellä. Opintojakson ensimmäinen toteutusajankohta on lukuvuonna 2013–2014, jolloin koostettu opintojaksomateriaali testataan kokonaisuudessaan. Insinöörityössä toteutettiin opetuskokeilu, jonka tarkoituksena oli arvioida ja testata opintojaksomateriaalin ja oppimistyökalujen toimivuutta käytännössä.

Opetuskokeilun onnistumisen ja kerätyn palautteen perusteella käytetty käänteisen luokkahuoneen malli ja opintojaksolla käytettävä opetusmateriaali soveltuvat ohjelmoinnin opetukseen ja oppimistyökalut, valittua lähiopetustyöskentelytyökalua lukuun ottamatta, soveltuvat hyvin web-ohjelmoinnin opintojaksolle. Opintojakson kehitystä voidaan viedä insinöörityön pohjalta eteenpäin.

Avainsanat	käänteinen luokkahuone, opetustyökalut, ohjelmoinnin opetus

Author	Maltari Minna
Title	New model to implement web-programming course
Number of Pages	46 pages + 1 appendices
Date	15 May 2013
Degree	Bachelor of Engineering
Degree Programme	Media engineering
Specialisation option	Digital media
Instructor	Principal Lecturer Kari Aaltonen

The aim of the thesis was to implement a web-programming course using the new teaching models and to improve the teaching of programming. The course was carried out using the flipped classroom method in which the student learns the theory part of the studies with the help of video tutorials and exercises. Classroom learning lessons focused on actively practicing the learned through many exercises and questions made by the teacher.

Suitable video tutorial material to replace theoretical lectures for this course was introduced. Also, a teaching platform with learning materials and exercises was compiled and a possibility to share the material online was created. A tool for classroom learning was also searched and tested.

The web-programming course consists of 14 study weeks and includes teaching of the basics of programming and object-oriented programming using the programming language Python. The course will be implemented the first time on forthcoming academic year 2013—2014, when the course material can be tested as a whole. A teaching experiment was carried out and it was designed to evaluate and test the course material and learning tools in practice.

Based on the results of the teaching experiment and feedback from the students the flipped classroom model and the used course teaching materials were found to be suitable for teaching programming. Also the learning tools, with the exception of the selected classroom teaching tool, were found useful. Course development can continue based on these project results.

Keywords	flipped classroom, teaching tools

Sisällys

1 Johdanto 1

2 E-oppiminen 3

2.1 Opetus ja verkko-oppiminen 3
2.2 Sulautuva oppiminen 4
2.3 Käänteinen luokkahuone 5

3 Web-ohjelmoinnin opintojakson toteutus 9

3.1 Suunnittelu 10
3.2 Opetussisältö 14

4 Oppimisympäristöt 17

4.1 Avoin verkkokurssimuoto MOOC 17
4.2 Koulutusteknologiat 19
4.3 Yhteisöllisiä työkaluja 21
4.4 Python-ohjelmoinnin opetus 22

5 Web-ohjelmoinnin opintojakson koostaminen 23

5.1 Valitut työkalut ja opetusmateriaalit 23
5.2 Tekninen toteutus 32
5.3 Testaus 37

6 Yhteenveto 41

Lähteet 43

Liitteet

Liite 1. Opetuskokeilun palautelomake

1 Johdanto

Tieto- ja viestintäteknologian kehittyminen ja uusien, erilaisten viestimistekniikoiden lisääntyminen ja yleistyminen ovat tuoneet paljon uusia mahdollisuuksia myös opetukseen ja oppimisympäristöihin. Insinöörityön tavoitteena on tutkia näitä uusia opetusmalleja ja hyödyntää niitä web-ohjelmoinnin perusteet -opintojakson suunnittelussa ja oppimateriaalin koostamisessa. Lähtökohtana on kehittää opetusta opiskelijalähtöisemmäksi ja siirtyä pois luentomuotoisesta opetusmallista. Insinöörityö tehdään Metropolia Ammattikorkeakoululle osana Smart Campus Metropolia -hanketta.

Insinöörityössä keskitytään erityisesti käänteisen luokkahuoneen opetusmalliin, jossa tavoitteena on tuntityöskentelyn aktivoiminen ja yksilölliseen opiskeluun kannustaminen. Opintojakson teoriaosuuksien opetus painottuu opiskelijan itsenäiseen opiskeluun videotutoriaalimateriaalin avulla. Yhteinen tuntityöskentely keskittyy opitun materiaalin omaksumiseen suoritettavien tehtävien avulla ja materiaalista mahdollisesti nousseiden kysymysten pohdintaan. Tässä opetusmuodossa opiskelijalla on mahdollisuus teorian sisäistämiseen omalla aikataulullaan, eikä eteneminen tai oppimistahti riipu muiden opiskelijoiden taitotasosta. Oppitunneilla on myös enemmän aikaa opiskelijan yksilölliseen ohjaamiseen ja vuorovaikutukseen opiskelijoiden kesken.

Insinöörityön projektiin sisältyy erilaisten tarjolla olevien videototeutuksien etsintä ja vertailu sekä niiden soveltuvuuden arviointi tämän opintojakson opetussisältöön. Tavoitteena on löytää kriteerit täyttävä opetusmateriaali, jonka pohjalta harjoitustyöt ja laboratoriotunnit koostetaan. Teoriavideoiden ja niihin liittyvien harjoitustöiden tueksi otetaan käyttöön jaettu verkkotyötila, jonka tavoitteena on luoda selkeä kanava opiskelijoiden ja opettajan väliseen vuorovaikutukseen, kuten kysymyksille ja opintojakson materiaalien jakoa varten. Myös lähiopetustunteja varten etsitään ja kokeillaan oppimista tukevaa yhteisöllistä verkkotyökalua, joka toimisi aktivoivassa osassa lähiopetuksessa. Oikeanlaisen työtilaratkaisun kriteereiden arvioiminen ja erilaisten ohjelmien toiminnallisuuksien testaus ja kokeilu opetusryhmän avustuksella on myös tärkeässä osassa insinöörityöprojektia.

Kehitettävän opintojakson opetussisältönä on ohjelmointikieli Python, ohjelmakehitysympäristönä Django ja sovelluskehitysalustana Google App Engine. Web-ohjelmoinnin perusteet -opintojakso on tarkoitettu ensimmäisen vuoden

mediatekniikan opiskelijoille, sen laajuus on 4 opintopistettä, ja sen ensimmäinen toteutusajankohta on suunniteltu alkavaksi lukuvuoden 2013—2014 keväällä. Opintojakso sisältää funktionaalisen ja jäsennellyn ohjelmoinnin perusperiaatteet ja periaatteiden soveltamisen yksinkertaisten oliopohjaisten verkkosovelluksien luomiseen. Opintojakso sisältää teoriaa, itsenäisiä harjoituksia ja lähiopetustunteja.

2 E-oppiminen

2.1 Opetus ja verkko-oppiminen

Termillä e-oppiminen viitataan erilaisten tieto- ja viestintäteknologioiden avulla toteutettuun verkko-oppimisen ja opetuksen muotoon. E-opetus voi koostua useilla eri tavoilla hyödynnetystä tietokoneavusteisesta opetuksesta, kuten internetin, intranetin tai erilaisten tietokoneohjelmien avulla toteutetusta etäopetuksesta, itseopiskelusta tai monimuoto-opetuksesta. [1.]

E-opetusta voidaan käyttää monella eri tavalla perinteisen opetuksen rinnalla. Sitä voidaan käyttää välineenä tiedon omaksumisessa ja soveltamisessa, tiedon haussa ja oppimisympäristöjen rakentamisessa. [1.]

E-opetusta on hyödynnetty yliopistotasolla jo lähes internetin alkuajoista lähtien. Ensimmäisiä internetin käyttäjiä olivat etenkin suomalaiset yliopistot, joiden yhteinen Funet-tietoverkko perustettiin jo vuonna 1984. E-oppimista on käytetty opetuksessa jo kymmeniä vuosia eri tietoverkkojen ja kokousjärjestelmien avulla sekä soveltaen myös tänä päivänä tunnettuja verkko-opetuksen muotoja kuten etä- ja monimuoto-opetusta. [2.]

Vaikka e-opetus on ollut olennainen osa koulutusjärjestelmää jo vuosikymmeniä, on vasta viime vuosina e-oppiminen ja verkko-opetus päässyt ottamaan suuria edistysaskelia sitten tietoverkkojen kehityksen ja näin hämmentänyt koko opetuskulttuuria. Syynä tähän edistykseen on informaatioteknologian yleistyminen ja entistä laajemmalle kehittynyt tietoverkkojen kasvu. Teknologia on sulautunut entistä tiiviimmin ympäristöömme ja sen käyttö on arkipäiväistynyt. Tämä ilmiö on kannustanut myös suurten yliopistojen opetusmuotojen uudelleen arvioimisen ja oppimista edistävien ratkaisujen kehittämisen vastaamaan tietoyhteiskunnan edistystä ja vaatimuksia sekä sen tarjoamia mahdollisuuksia. [2.]

Koulutusteknologian kehitys tuo uusia mahdollisuuksia toteuttaa opetus uudella ja tehokkaammalla tavalla. Moni nykyaikainen yliopisto on jo kehityksen myötä siirtynyt käyttämään opetuksen tukena erilaista uutta koulutusteknologiaa, ja opetuksen tavat ovat muuttuneet. Opetuksessa on aktiivisesti alettu miettiä ratkaisuja toteuttaa opetus

oppijalähtöisesti ja tukea yksilöllistä oppimista ja siirtyä pois opettajalähtöisestä luentomallista. [3.]

E-opetuksen määritelmä on hyvin laaja, ja sillä voidaan viitata moneen erilaiseen e-oppimismuotoon. Tässä insinöörityössä keskitytään e-oppimiseen, jonka määrittelee erityisesti sulautuva oppiminen, opiskelijakeskeinen oppiminen ja käänteisen luokkahuoneen malli.

2.2 Sulautuva oppiminen

Sulautuvan oppimisen näkökulma liittyy vahvasti e-oppimisessa jo kauan käytössä olleeseen monimuoto-opetukseen, jossa periaatteena on yhdistellä opetukseen erilaisia elementtejä, kuten ryhmä-, verkko- ja luento-, lähi- ja etäopetusta. Sulautuvan oppimisen periaatteena on oppimisympäristöjen sulauttaminen saumattomaksi kokonaisuudeksi, joka sisältää erilaisia opetusmenetelmiä ja suuntauksia. [4.]

Sulautuvassa oppimisen menetelmässä yhdistellään eri tavoilla kontakti- ja verkko-opetusta ja erilaisia pedagogisia ratkaisuja ja opetustekniikoita. Sulautuvan oppimisen tavoitteena on luoda monimuotoinen, oppimisprosesseja tukeva ja mielekäs oppimisympäristö. Määritelmän mukaan sulautuva oppiminen on oppimista ja opetusta, jossa hyödynnetään tieto- ja viestintäteknologian kehitystä perinteisen opetuksen tukena. [5.]

Sulautuminen mahdollistuu nykyajan teknistymisen avulla, jossa informaatiota on tarjolla opetuksen tueksi lukuisista eri kanavista, verkkoympäristöt ovat osa arkea, eikä oppimisprosessi rajoitu vain oppilaitosympäristöön. Opiskelijoiden on helppo toteuttaa oppimistaan ajasta ja paikasta riippumatta, ja oppiminen sulautuu osaksi arkielämää. [4.]

Tavallinen e-oppimisen määritelmä sisältää verkko-opetuksen mallin, joka saattaa keskittyä lähinnä vain tekstipohjaiseen tiedonjakoon, tehtävien antoon tai oppimistehtävien palautukseen oppimisalustojen avulla, eikä niinkään rikkaaseen oppimisympäristöön. Sulautuvassa opetuksessa päämääränä on erityisesti monimuotoinen oppimisympäristö, jossa tekniikkaa käytetään monipuolisemmin kuin vain opiskelijahallinnointiin. [5.]

Sulautuvan oppimisen näkökulma painottaa opiskelijakeskeistä näkökulmaa, jossa opiskelijalla on aktiivinen rooli. Sillä pyritään rakentamaan vuorovaikutteinen oppimisympäristö, joka kehittää oppimisprosesseja. Opettajan rooli sulautuvassa oppimisessa on enemmän ohjaajan ja asiantuntijan. [6.]

Teknologia sulautuu osaksi arkipäivää. Sulautuvaan oppimiseen vaikuttaa myös lisääntynyt tiedonjako verkossa. Sähköinen tietomäärä ylittää painettujen kirjojen tietomäärän, ja tekniikka sulautuu osaksi kaikkia toimintoja.

Insinöörityössä keskityn sulautuvan oppimisen malliin, jossa opetuksen elementtejä on yhdistetty niin, että lähiopetuksen lisäksi on videotutoriaalimateriaalia. Oppimisprosessissa yhdistetään sekä itsenäistä että ryhmämuotoista oppimista, niin kontaktiopetuksen kuin verkkoyhteisönkin muodossa. [7.]

2.3 Käänteinen luokkahuone

Käänteisen luokkahuoneen malli on osakokonaisuus sulautuvan oppimisen periaatetta. Mallin mukaisesti käänteisen luokkahuoneen osa oppimisympäristöistä sulautuu osaksi ympäristöä, eikä ole laite- tai paikkasidonnaista. [7.]

Käänteisen luokkahuoneen opetusmalli on suhteellisen uusi ilmiö, sillä erilaisten videoiden julkaisupalveluiden suosio ja yleistyneisyys sekä vuorovaikutuskanavien kehittyminen on vaikuttanut vahvasti opetustyyleihin ja näin myös käänteisen luokkahuoneen mallin kehittymiseen. [8.]

Käänteisen luokkahuoneen opetusmalli tarkoittaa sitä, että opetuksen työnkulku on käännetty päinvastaiseksi. Yleisesti opetus perustuu siihen, että opettaja pitää luennon opiskelijoille ja kertoo käsiteltävän aiheen sisällön, joka opiskelijoiden on tarkoitus omaksua, ja oppitunnin päätteeksi opettaja määrää tietyn verran harjoituksia suoritettaviksi kotona itsenäisesti. Käänteisen luokkahuoneen mallissa opiskelijalle on jo ennen tunnin alkua määrätty tietty, sen opetuskerran sisältöä koskeva materiaali omaksuttavaksi ja aikaisemman mallin mukaiset niin kutsutut kotitehtävät tehdään yhdessä lähiopetustunnin aikana. [8.]

Opiskelija voi perehtyä materiaaliin ennen oppituntia omalla tahdillaan ja omaehtoisessa aikataulussaan, ja lähiopetuskerralla materiaalia tarkastellaan yhdessä muiden opiskelijoiden ja opettajan kanssa sekä käydään läpi mahdollisia ongelmakohtia ja materiaaliin liittyviä kysymyksiä. [8.]

Käänteisen luokkahuoneen mallin yleistymisistä on edesauttanut KhanAcademyssä sovellettu opetustapa. Salman Khan aloitti toimintansa lataamalla videopalvelu Youtubeen matematiikan opetusvideoita serkkujaan varten. Hetkessä muutkin käyttäjät huomasivat videoiden kiinnostavuuden ja havaitsivat mahdollisuuden sisäistää asioita aivan uudella tavalla. Opiskelijat huomasivat myös oppimismuodon positiivisen vaikutuksen oppimiseen verrattuna luentopohjaiseen opetustapaan. [8.]

Videomuotoinen teoriaopetus on hyödyllinen opiskelijan kannalta monella tapaa. Opiskelijalla on mahdollisuus tutustua materiaaliin omaan aikatauluun sovitetusti ennen oppituntia. Tämä mahdollistaa opiskelijalle vapauden valita opiskeluajankohtansa ja paikkansa. Opiskelusta tulee omaehtoista, eikä pakollisen läsnäolon vaativaa luentomuotoista opetusta. Videomateriaaliin voi myös perehtyä osissa, jos se opiskelijan omaksumistahdin kannalta on parempi ratkaisu. [8.]

Luento-opetus ei myöskään usein pysty takaamaan täyttä asian omaksumista koko opiskelijaryhmälle. Opiskelijoiden keskinäinen lähtötaso opintojakson sisältöön nähden voi olla hyvinkin erilainen. Tämä luo haasteita opetusvauhtiin, sillä kaikkia yksilöitä ei pystytä ottamaan huomioon asioiden käsittelyn etenemistahdissa. Videoteoriaopetus antaa opiskelijoille mahdollisuuden käydä opetettava aihe läpi omassa tahdissaan, katsoa uudelleen osioita, joihin tarvitsee enemmän aikaa omaksua, palata videossa kohtiin, joihin viitataan myöhemmässä materiaalissa, ja kerrata niitä tai jopa harpata yli osioita, jotka ovat jo entuudestaan tuttuja ja jo omaksuttuja. Opiskelija voi keskittyä oman oppimisensa kannalta tärkeisiin seikkoihin ja asioihin, jotka kiinnostavat. Materiaaliin on myös helppo palata jälkeenpäin, jos mieleen on jäänyt jokin osio, jota haluaa kerrata. [8.]

Kun teoriaosuus omaksutaan ennen yhteistä oppituntia, voidaan itse lähiopetustunnilla keskittyä keskustelemaan ja kertaamaan asioita, jotka ovat jo tuttuja. Lähiopetusaika voidaan käyttää näin paremmin hyödyksi erilaisten tehtävien ja ryhmätöiden avulla. Näin voidaan myös kerrata opittua ja parantaa oppimisprosesseja. [8.]

Kun lähiopetusaika on varattu tehtäville ja yhdessä oppimiselle, on opettajallakin enemmän aikaa panostaa yksilöihin, jotka tarvitsevat enemmän ohjeistusta.

Flipped classroom eli käänteinen opetus voidaan määritellä osakokonaisuudeksi yksilöllisen oppimisen opetusmenetelmää. Yksilöllisen oppimisen menetelmä yhdistelee muun muassa aikaisemmin mainittua sulautuvan oppimisen mallia ja käänteisen luokkahuoneen mallia opetusmenetelmänään. Yksilöllisen oppimisen opetusmallissa voi oppiminen olla yhteisöllistä tai itsenäistä, mutta pääpaino on opiskelijan omaan tiedonrakennusprosessiin keskittynyttä. Opiskelija rakentaa osaamistaan, ongelmanratkaisuaan ja ymmärtämistään yksilöllisenä toimijana. [7.]

Luennoista luovuttaessa itse lähiopetustilanne muuttuu, ja on tärkeää kiinnittää huomiota siihen, että opiskelijoilla on opintojakson materiaalin omaksumiseen samalla tavalla mahdollisuus kuin aikaisemmassa mallissa. Luennoinnista on luovuttu myös Peer Instruction (PI) -mallissa, jota on käytetty onnistuneesti fysiikan opetuksessa. Kun luovutaan luennoista, on tärkeää kiinnittää huomioita tuntityöskentelyyn, jota PI:n mallissa painotetaan. [9.]

Peer Instruction on opetuksessa käytettävä vertaisohjausmalli, jossa tavoitteena on käänteisen luokkahuoneen opetusmallin menetelmiä hyödyntäen parantaa opiskelijoiden ja opettajan välistä vuorovaikutusta oppimisen edesauttajana. Peer Instruction opetuksessa esittävistä luennoista on luovuttu ja oppiminen tapahtuu lähiopetustunneilla keskustelemalla aihealueesta opiskelijoiden itsenäisesti omaksutun tiedon pohjalta, ja näin tietämys aiheesta syventyy. [9.]

Siirryttäessä Peer Instruction oppimismalliin perinteisestä luennoinnista opettajan rooli erityisesti tiedon soveltamisen ja opitun ymmärtämisen herättelijänä kasvaa. Opettaja siirtyy luennoitsijasta enemmänkin ohjaajan tai tutorin rooliin ja pääpaino on opiskelijoissa, jotka saavat tilaisuuden omaksua tietoa. [9.]

Peer Instructionin kehitti alun perin Harvardin yliopiston opettajan Eric Mazur parantamaan oppimista yliopisto-opetuksessa. Oppitunnin työnkulkuprosessi etenee niin, että opettaja esittää kysymyksen pohjautuen opiskelijoiden ennen oppituntia omaksumaan aiheeseen ja jokainen opiskelija pohtii vastausta tahollaan. Vastauksiin pohjautuen keskustellaan pienryhmissä tai pareittain vertaiskeskustelun muodossa ja jokainen muotoilee vastauksensa uudelleen. Lopuksi kysymystä pohditaan vielä

opettajan johdolla koko luokan kesken. Opettajan esittämät kysymykset ovat usein monivalintakysymyksiä, mutta muunkin tyyppisiä yksinkertaisia kysymyksenasetteluja voi käyttää. [9.]

Suomessa on mallia käytetty onnistuneesti muun muassa fysiikan opetuksessa. Jyväskylän yliopiston fysiikan laitoksen mekaniikan perusopintojen opintojaksolla oppiminen on toteutettu Peer Instruction mallin mukaisesti akatemiatutkijan Pekka Koskisen johdolla. Koskisen kurssilla on luovuttu perinteisestä luentomallista ja opiskelijoiden vuorovaikutusta on lisätty. Opiskelijoilta edellytetään ennen lähiopetustunnille osallistumista perehtymään aiheisiin oppikirjan avulla. Opettaja ei käytä luentoaikaa vain teorian läpikäyntiin, vaan alun lyhyen kertauksen jälkeen luento pohjautuu keskustelun johdatteluun ja opiskelijoiden omaan pohdintaan ja keskinäiseen vuorovaikutukseen. Opettajan rooli on esittää kysymyksiä tunnin aiheista ja sen avulla herätellä opiskelijoiden omaa kykyä muovata oppimista ja argumentointia. Erilaisten kysymysten esittäminen onkin Peer Instruction mallissa avainasemassa, jotta tunnilla pystytään paneutumaan opittaviin käsitteisiin ja pitämään oppimisen mielenkiinto yllä. [9.]

Malli on tuonut hyviä tuloksia. On huomionarvoista painottaa opiskelijoille materiaalin läpikäymistä ennen tunteja, jotta tunneilla aikaa ei kulu turhaan uudelleen asioiden läpikäymiseen. Opiskelijatkin näin huomaavat itsenäisesti tehdyn opiskelun hyödyllisyyden ja saavat lähiopetuksesta enemmän irti. Niin kuin yleensäkin, muutokseen suhtautuminen on ensin varovaista ja opiskelijoiden suhtautuminen aktiiviseen opiskeluun on aluksi hieman epäileväistä. Pian kuitenkin suhtautuminen muuttuu myönteisemmäksi ja uusi menetelmä omaksutaan harkinnan jälkeen hyvin. [9]

Peer Instruction on opetusmalli, jota on käytetty onnistuneesti myös tietojenkäsittelyn opetuksessa, ja se toimisi hyvin ohjelmoinninopetuksessakin. Opetusmallia on käytetty useasti muissakin oppiaineissa, kuten matematiikassa ja tulokset ovat olleet hyviä. [9.]

Ohjelmoinnin opetuksen pedagogiikka

Ohjelmoinnin opetuksessa e-oppimisen monimuoto-opetusta on käytetty jo kauan, sillä aihealue on hyvin käytännönläheinen, ja parhaat tulokset opetuksessa voidaan saavuttaa sekä tietokoneavusteisuutta että lähiopetusta yhdistellen. Opetustyylit

vaihtelevat kuitenkin jonkin verran, ja opintojaksojen toteutustyyli on usein opettaja- eikä niinkään oppijalähtöinen.

Ohjelmoinnin opiskelu ja sen taito on hyvin moniosainen ja vaatii keskittynyttä harjaannuttamista. Ohjelmoinnin oppimisprosessissa tärkeä näkökulma opetuksen kannalta on tarjota taidot ohjelmointikielen erilaisten rakenteellisten ominaisuuksien hallitaan ja taito yhdistää niitä sopiviin kokonaisuuksiin. Pelkästään mekaaninen ohjelmointikielen syntaksin ja semantiikan hallitseminen ei riitä. [10.]

Ohjelmoinnin opetuksessa käytetään usein pedagogista mallia nimeltä Cognitive Apprenticeship. Helsingin yliopistossa tietojenkäsittelytieteen laitoksella on toteutettu useita ohjelmoinnin kursseja käyttämällä tehostetun kisällioppimisen nimistä pedagogista mallia, joka on kehitetty laitoksella tehostamaan oppimista. Malli pohjautuu Cognitive Apprenticeshipiin, ja mallin avulla tehostetaan ohjelmoinnin ja tietojenkäsittelyn kurssien opetustuloksia. Kisällioppimisessa tärkeimpänä opetusmenetelmänä on opiskelijan oma oivallus ja kokeilemalla tekeminen, riittävän ohjauksen avulla. [10.]

Tässä insinöörityössä sovelletaan tehostetun kisällioppimisen mallista tuttuja pedagogisia malleja ja hyödynnetään sen tuottamia aikaisempia tuloksia tuomaan lisätietoa toteuttaa web-ohjelmoinnin opintojakso toimivaksi kokonaisuudeksi.

3 Web-ohjelmoinnin opintojakson toteutus

Ohjelmoinnin opetus liittyy olennaisena osana mediatekniikan opintoihin ja kuuluu kiinteänä osa-alueena media-alaan ja siihen tähtääviin opintoihin. Taito hallita ohjelmoinnin perusteet, myöhemmistä koulutuksellisista ja ammatillisista valinnoista riippumatta, luo hyvät tietotaidot monelle eri mediaan liittyvälle alalle. Ohjelmoinnin perusteisiin ja olio-ohjelmointiin perehdytetään jo opiskelijoiden ensimmäisen opintovuoden aikana perusopinnoissa oleellisena osana välttämättömiä opintoja. Opintojaksolla käsiteltävät aiheet ja niiden sisäistäminen liittyvät merkittävästi myös kykyyn omaksua myöhempien ammatillisten opintokokonaisuuksien sisältöä, kuten yhteisiin ammattiopintoihin kuluva tiedonhallinta ja julkaisutekniikka sekä

tietojenkäsittely ja viestintätekniikka. Jos opiskelija suuntautuu digitaalisen median opintoihin, ovat ohjelmoinnin taidot huomattavassa osassa.

Media-alalla suuntaus ohjelmoinnin ja tietokoneavusteisiin tehtäviin on jatkuvasti kasvussa, ja sovellusten ymmärtäminen ja taitojen soveltaminen tuo huomattavaa etua työmarkkinoilla. Ohjelmoinnin taitajista on jatkuva kysyntä, joten opetuksen tärkeyttä ei voi tarpeeksi korostaa. On tärkeää, että ohjelmoinnin opintojakso opettaa opiskelijalle taidot ohjelmoinnin perusrakenteista ja vuorovaikutteisista sovelluksista. Tärkeintä ei siis ole vain tietyn ohjelmointikielen ominaisuuksien ja dynaamisten toimintojen opiskelu, vaan yleinen ohjelmoinnin ymmärtäminen ja johdatus sen maailmaan.

Ohjelmoinnin opetussisältönä mediatekniikan koulutusohjelmassa edeltävinä vuosina on käytetty ohjelmointikieltä PHP. Aikaisempina vuosina ohjelmoinnin opetus on jaettu kahteen opintojaksoon, ohjelmoinnin perusteet 2 op olio-ohjelmointi 2 op. Opetus koostui PHP:n opiskelusta, ohjelmointityökaluna oli Eclipse ja virtuaaliympäristönä xampp. Opintojaksokokonaisuus oli rakennettu niin, että viikko-ohjelma koostui kahden oppitunnin mittaisista luento-osuuksista ja tietokoneluokassa tehdyistä opettajan johdolla suoritetuista harjoitustehtävistä ja viikoittain palautettavista kotitehtävistä. PHP on Pythonin tavoin olio-orientoitunut ohjelmointikieli, ja sen avulla on helppo opettaa rakenteellista ohjelmointia. Sisältönä aikaisemmalla opintojaksolla oli myös html, sillä pelkän PHP:n avulla ei voi muodostaa web-sovelluksia. Tietokantojen rakentaminen ja hallinnointi tapahtui MySQL:n avulla.

Opetussisältö muokkautuu luonnollisesti markkinoiden ja kysynnän ja uusien suuntauksien mukaan. Ohjelmoinnin opetussisältökin vaihtuu tietyin välein sen perusteella, mikä ohjelmointikieli nähdään tarpeellisimmaksi ja hyödyllisimmäksi opiskelijan kannalta osata. Vuosia sitten oli Java, josta siirryttiin PHP:hen ja nyt luonnollisena jatkumona Pythoniin.

3.1 Suunnittelu

Web-ohjelmoinnin opintojakson ohjelmointikielenä on Python 2.7, ohjelmistokehyksenä Django ja ohjelmointiympäristönä pilvipalveluna Google App Engine - ohjelmointityökalu.

Totutettavan opintojakson nimi on Web-ohjelmoinnin perusteet, ja sen laajuus on 4 op, yhteensä 14 opiskeluviikkoa. Tulevassa toteutuksessa aikaisemmin kahteen eri kurssiin hajautetut aihealueet on kuitenkin sulautettu yhteen muodostamaan tiiviimpi kokonaisuus ja jatkuvuus ohjelmoinnin opetukseen.

Uuden mallin mukaisen web-ohjelmoinnin perusteet -opintojakson ensimmäinen toteutusajankohta on suunniteltu alkavaksi lukuvuoden 2013—2014 keväällä, kolmannella periodilla. Opintojakso on opetussuunnitelman mukaisesti suunnattu ensimmäisen vuoden opiskelijoille.

Opetus toteutetaan aikaisemmin järjestettyjen ohjelmointiopintojaksojen toteutustavasta poiketen. Uudenmuotoista opetusmallia ryhdyttiin kehittämään lähtökohdasta, jossa haluttiin parantaa oppimista ohjelmoinnin opetuksessa. Aikaisempina vuosina on havaittu teorialuentojen käyneen melko tehottomiksi ohjelmoinnintaitojen oppimisessa ja sisäistämisessä. Käytännönläheisen aiheen, kuten ohjelmoinnin, opetuksessa lähestymistapana toimii usein parhaiten kokemuksien kautta omaksuminen. Siksi opettajalähtöiset teorialuennoinnit aiheesta eivät usein pysty välittämään opiskelijalle tietotaitoja parhaalla mahdollisella tavalla.

Opintojaksoa lähdettiin rakentamaan näkökulmasta, jossa opetus olisi helposti omaksuttavaa lähtötasoltaan hyvin erilaisille opiskelijoille. Oppimisympäristö muuttuisi aikaisemmasta mallista ja käyttöön otetaan oppimista helpottavia oppimistyökaluja, jotka mahdollistavat sulautuvan ja yksilöllisen oppimisen opetusmallit.

Käänteisen luokkahuoneen mukaisesti läsnäolopakolliset teorialuennot korvataan videotutoriaalien avulla. Tarkoituksena oli löytää videomateriaali, jonka avulla ohjelmoinnin teoria välitetään opiskelijalle selkeällä ja havainnollistavalla tavalla. Tarkoituksena ei ole vain siirtää luennointia verkkoon, vaan selkeyttää opetusta ja tarjota opetusmateriaali tiiviinä kokonaisuutena, jonka pohjalta on helppo soveltaa opittuja asioita tehtävien parissa.

Opintojaksolle osallistuu opiskelijoita, joilla on erilainen lähtötaso ohjelmoinnin opiskeluun. Osalla saattaa olla jo jonkinlaista kokemusta ja kiinnostusta ohjelmoinnista ja siksi hän onkin hakeutunut tekniselle alalle, osalla lähtötaso on väljempi. Videotutoriaalien avulla opiskelijat voivat keksittyä aiheisiin, jotka kiinnostavat. Paremmilla ohjelmoijilla on varaa keskittyä uusiin asioihin ja jättää tutut aiheet

vähemmälle huomiolle. Näin vältytään luennoilla turhautumiselta. Myös hitaammat opiskelijat voivat siirtää takaisinpäin videon katselukohtia ja katsoa vaikeat kohdat läpi aina uudelleen. Käänteinen luokkahuone ei vaadi läsnäoloa, joten opiskelijat, jotka eivät pääse luennoille, pysyvät silti mukana opetustahdissa.

Laboratoriotuntityöskentely rakennetaan videotutoriaalimateriaalin pohjalta aiheen omaksumiseen käytännön ja soveltamisen kautta. Lähiopetustunnit koostuvat erilaisista harjoitustehtävistä, jotka tehdään opettajan johdolla tunnin aikana. Tunneille laaditaan tehtävät, jotka kertaavat jo opittuja asioita, mutta eivät ole samoja kuin videolla esitetyt esimerkit. Tavoitteena on saada opiskelijat hyödyntämään oppimaansa ja ymmärtämään keskeiset ohjelmoinnin ominaisuudet.

Tunti etenee harjoitustehtävien annon mukaan. Tehtävät ovat koko luokalle samat, mutta yhtenäisen tehtävänannon avulla eritasoiset oppijat voivat edetä omaan tahtiinsa ja opettaja toimii ohjeistajana ja tarjoaa apuaan kierrellen väillä luokassa. Näin opettaja saa hyvän käsityksen luokan etenemisvauhdista ja voi tarjota apuaan sitä eniten tarvitseville. Tunnilla sovelletaan myös xa:n mukaista oppimista, jossa opettaja on ohjaavassa roolissa ja pääpainopiste on tehtävillä ja itse tekemisellä.

Oppitunnin aikana opettaja aktivoi opiskelijoita PI-opetusmallin mukaisesti erilaisilla yksinkertaisilla kysymyksillä aihealueesta. Keskeisiä ohjelmoinnin aiheita käydään läpi opettajan esittämien kysymysten avulla, joihin jokainen osallistuja vastaa itsenäisesti lähettämällä vastauksen kaikkien näkyville tunnilla käytettävän yhteisötyökalun avulla. Vastausten pohjalta opettaja voi johdatella opiskelijoita keskustelemaan aiheesta keskenään ja yleisellä tasolla. Vastausten avulla opettaja huomioi, oliko aihe omaksuttu, ja keskustelun johdattelemin lisäkysymyksin opiskelijat muokkaavat vielä vastauksiaan uudemman kerran. Kun opiskelijat ovat sisäistäneet oikean vastauksen, voidaan siirtyä eteenpäin.

Toteutus

Opintojakson toteuttamiseksi etsittiin sopiva oppimisalusta. Kriteereitä olivat helppo käyttöliittymä, keskustelualusta ja mahdollisuus jakaa materiaalia opettajan ja opiskelijoiden kesken.

Keskeiset kriteerit videotutoriaalimuotoiselle opetusmateriaalille olivat sisällön selkeys ja tasapainoisuus. Tärkeä seikka audiovisuaalisissa toteutuksissa on myös äänen- ja kuvanlaatu. Opetuksen etenemisvauhdin on myös oltava tarpeeksi joutuisa ja dynaaminen.

Kuten useissa Mooc-toteutuksissa on todettu, parhaat oppimistulokset saadaan aikaan yhdistettäessä videoluennot harjoitustehtävien kanssa. Aluksi videotutorialien seuraksi suunniteltiin pienimuotoisia kirjallisia monivalintatehtäviä, joiden avulla varmistettaisiin, että opiskelija on katsonut ja omaksunut tutoriaalin asiat. Parhaaksi vaihtoehdoksi kuitenkin havaittiin videomateriaaliin liittyvät harjoitustehtävät, joiden tarkoitus on auttaa asian omaksumista eikä niinkään vain varmistaa, että opiskelija on katsonut videon. [11.]

Insinöörityön tarkoituksena oli myös valita sopiva tuntityöskentelytyökalu helpottamaan laboratoriotuntityöskentelyä. Tarkoitukseen sopiva työkalu haluttiin löytää helpottamaan tiedonjakoa tunnin aikana, kuten esimerkiksi linkkien ja tehtävänantojen helppoa saatavuutta. Myös ongelmatilanteiden ratkaisuun ja opiskelijoiden esittämien kysymysten ratkaisemiseen oli yhteisöllisen työkalun avulla tavoitteena helpottaa työskentelyä. Usein monista opiskelijoista koostuvilla ryhmätunneilla sama ongelmatilanne esimerkiksi koodin kanssa toistuu usealla opiskelijalla. Työkalun avulla on tarkoitus helpottaa työskentelyä, kun opettaja pystyy auttamaan useampaa opiskelijaa kerrallaan esimerkiksi laittamalla ongelman ratkaisun näkyviin kaikille yhteisötyökalun avulla tai vastaamalla alustalle lähetettyyn kysymykseen, josta se on helppo kaikkien lukea. Työkalun avulla on myös mahdollista soveltaa niin sanottua peer instruction -mallia, jossa opiskelijat voivat tallentaa vastauksensa yhteisöalustalle ja oppia muiden vastauksista. Opettaja saa myös työkalun avulla lähetettyjen vastausten avulla kattavan käsityksen opiskelijaryhmän aiheen omaksumisesta. Helposti opettajan esittämät kysymykset jäävät opiskelijoille epäselviksi tai niiden pohtiminen jätetään vain aktiivisimpien opiskelijoiden vastuulle. Tuntityökalun avulla jokainen opiskelija voi osallistua aktiivisesti. Työkalu madaltaa kynnystä osallistua opetukseen opiskelijoilla, jotka tavallisesti epäröisivät vastaamista koko luokan kuullen.

Ohjelmoinnin opetuksessa tunnilla ohjelmoimisessa käytetään ohjelmointityökalua Google App Engine, jota on helppo käyttää ja myös helppo opiskelijoiden asentaa kotikoneelle ja toistaa tehtäviä, joita tunnilla on käsitelty.

3.2 Opetussisältö

Oppimisympäristö osana Smart Campus -projektia

Smart Campus Metropolia on Euroopan unionin rahoittama projekti, joka kuuluu osana Competitiveness and Innovation Framework Programme -puiteohjelmaan. Projekti lähti käyntiin syksyllä 2012, ja se on kestoltaan yhteensä 2,5 vuotta. Smart Campus-hankkeessa ovat Metropoliasta mukana Leppävaaran ja Myyrmäen toimipisteet. [12.]

Tavoitteena Smart Campus -hankkeessa on kehittää erilaisia kampusalueen toimintoja ja palveluita, joilla tarjotaan hyvinvointia, viihtyvyyttä ja jaksamista koko henkilökunnalle ja opiskelijoille. Projektin avulla parannetaan energiatehokkuutta, palvelutarjontaa ja oppimisympäristöjä sekä erilaisia toimintoja, kuten terveys-, ruoka-, opiskelu- ja sosiaalipalveluja. Projektin avulla parannetaan myös energiatehokkuutta ja edesautetaan muutosta entistä energiaystävällisempään kulutustottumusten muutokseen. [12.]

Kampusten kehittämisessä hyödynnetään erilaisia tietoteknisiä ratkaisuja, langatonta ja ekoteknologiaa, joilla vaikutetaan palveluihin, toimitiloihin sekä opetukseen ja oppimiseen[12].

Tämä insinöörityö liittyy Smart Campus Metropolia -projektiin oppimisympäristöjen sekä opetuksen ja oppimisen kehittämisenä. Web-ohjelmoinnin perusteiden opintojakson uusien, oppimisen kannalta tärkeiden näkökulmien tarkastelu sekä oppimisympäristön kokeilu ja rakentaminen toimivat projektin kannalta oppimisen ja oppimisympäristöjen kehittämisenä.

Ohjelmointikielenä Python

Python on yleisesti käytetty ohjelmointikieli. Sitä käytetään sekä web-sovellusten että työpöytäsovellusten ohjelmointiin. Se on käyttötarkoituksiltaan monimuotoinen, eikä se ole tavoitteellisesti erikoistunut mihinkään tiettyyn tehtävään. Tämän vuoksi sitä käytetään laajasti eri ohjelmointisovelluksissa ja web-kehityksessä. Python on tyyliltään verrattavissa sellaisiin ohjelmointikieliin kuin Perl, Ruby ja Java. [13.]

Pythonissa on laaja sisäänrakennettu standardikirjasto, joka tarjoaa kaiken tarvittavan jopa monimutkaistenkin sovellusten kehittämiseen, kuten esimerkiksi pelikehitykseen, graafisiin käyttöliittymiin ja web-ohjelmistokehitykseen. [13.]

Yleisesti Python on melko tunnettu ja käytetty ohjelmointikieli. Se on kahdeksantena TIOBE-ohjelmointiyhteisön toukokuussa 2013 tekemässä arvostelussa ja säilyttänyt asemansa listalla jo useamman vuoden. TIOBE-ohjelmointiyhteisön arvostelu eri ohjelmointikielien suosiosta perustuu maailmanlaajuisiin lukuihin eri ohjelmointikielien taitajista ja niiden suosiosta erilaisten opetusmateriaalien ja käyttäjien keskuudessa. [14.]

Ohjelmontikielenä Python on syntaksiltaan helppolukuinen. Ohjelmointikielessä termillä syntaksi tarkoitetaan merkistön säännöstöä ja muotoa. Ohjelmoinnissa koodin kirjoitusasun on oltava hyvin formaali ja tarkkaan määritelty, jotta tietokoneohjelma pystyy lukemaan ja tulkitsemaan oikein, mitä toimintoja tai lopputuloksia kirjoitetulla koodilla yritetään saavuttaa. Pythonin syntaksi on muihin yleisiin ohjelmointikieliin verrattuna selkeämpi ja tiiviimpi. Esimerkiksi erilaiset koodilohkot määritellään sisennysten ja väli-lyöntien perusteella eikä aaltosulkeita ja puolipisteitä käyttämällä, kuten yleisesti on tapana ohjelmointikielissä. Näin koodi on visuaalisesti itse ohjelmoijalle ja koodin tulkitsijalle selkeä tarkkaan määritellyn ja muuttumattoman rivijaottelun avulla. Python on suunniteltu nopeaksi ja helpoksi oppia, käyttää ja ymmärtää. [15; 13.]

Python on dynaamisesti tyypittävä ohjelmointikieli, eli muuttujan tyyppi määritellään sille sijoitetun entiteetin arvon perusteella. Toisin kuin esimerkiksi C, C++ tai Java, jotka ovat staattisesti tyypittäviä, Python-ohjelmointikielessä muuttujien helppojen tyyppi-muunnosten avulla kehitys on nopeampaa ja ketterämpää. [13.]

Python on ohjelmointiparadigmaltaan olio-orientoitunut ohjelmointikieli, jonka käyttää olioiden luokkapohjaista perintää. Perimisessä oliot ovat hierarkkisessa rakenteellisessa yhteydessä toisiinsa määriteltyjen luokkien avulla. Luokat voivat periä ominaisuuksia ja muuttujia muilta ylemmiltä luokilta ja käyttää uudelleen niiden ominaisuuksia, joutumatta määrittelemään näitä uudelleen. [13.]

Olio-orientoituneen ohjelmointikielen hyöty ohjelmoinnissa ja ohjelmoinnin opetuksessa ovat sen helppo hallittavuus ja koodin uudelleenkäytettävyys. Ajatusmallina olio-

ohjelmoinnissa on, että jo kertaalleen luotuja olioita voi helposti käyttää uudelleen eri tarkoituksiin ja sovellus kootaan pienistä palasista ja funktioista, jotka pystytään tarkoin määrittelemään ja jotka ovat helposti hahmotettavissa. [16.]

Olio-ohjelmoinnissa koostettava sovellus kuvataan olioina, joilla on erilaisia ominaisuuksia ja toimintoja ja niiden välisiä suhteita, eli ne vaikuttavat näin muiden olioiden käyttäytymiseen [16].

Python on ollut käytössä jo kauan, ja ohjelmointikielen versioita on monia. Python-ohjelmointikielen on kehittänyt Guido van Rossum, ja Python 2 julkaistiin vuonna 2000, jolloin kehitys sai enemmän yhteisöllistä tukea. Python 3 on jo julkaistu, mutta se ei ole silti vielä korvannut käytössä Pythonin aikaisempaa versoita. Python 3 eroaa versiosta 2 vain hieman syntaksiltaan. Python on melko vanha ohjelmointikieli, ja sitä on kehitetty kauan suuren vapaaehtoisyhteisön voimin. [13.]

Pythonin ohjelmointikehityksen tueksi valittiin ohjelmistokehykseksi Django, jota kuvataan seuraavassa luvussa.

Ohjelmistokehyksenä Django

Django on vapaan lähdekodin ohjelmistokehysalusta eli työkalu web-sovellusten rakentamiseen. Alun perin Django on kehitetty World Online -sanomalehden verkko-osaston työkaluksi. Kun web-kehityskielestä PHP siirryttiin käyttämään Pythonia vuonna 2003, Adrian Holovaty ja Simon Willson kehittivät Djangon interaktiivisempaa ja nopeampaa web-kehitystä varten. [17.]

Nykyään Djangoa ylläpitää vapaaehtoisista koostuva voittoa tavoittelematon ja itsenäinen Django Software Foundation (DSF). Django on toteutettu Python-ohjelmointikielellä, ja sen keskeinen tarkoitus on tarjota yksinkertaista ja helppoa ohjelmistokehitystä monitasoisten ja tietokantapohjaisten web-sivujen luomiseen. [17.]

Django on alun perin suunniteltu nopeatempoisten uutisten julkaisuympäristöksi. Tärkeässä asemassa sen kehityksessä olivat nopean kehityksen ja puhtaan käytännöllisen muotoilun ja ulkoasun vaatimukset. Django mahdollistaa korkean suorituskyvyn ja selkeän ulkoasun omaavien web-sovellusten kehityksen nopeallakin aikataululla. [17.]

Djangon avulla voi rakentaa tietokantapohjaisia, interaktiivisia sovelluksia, jotka toimivat selaimessa. Django tarjoaa tämän rakenteen ja yleiset metodit sovelluksen kokoamiseen. Sitä käytetään helpottamaan tietokantapohjaisten www-sivujen koostamista. Näitä ominaisuuksia muokataan Pythonin avulla. Djangon avulla voidaan myös hallinnoida käyttöliittymän muokkausta admin-tasolla. [17.]

Django on MVC-sovelluskehys eli Model-view-controller-pohjaa noudattava ohjemisto-arkkitehtuuri. Tässä teknisessä arkkitehtuurissa ohjelma jaotellaan kolmeen järjestelmän toiminnan kerrokseen eli malliin, näkymään ja käsittelijään. Näistä ensin mainittu eli model vastaa kokonaisuudessa liiketoimintalogiikasta, view näkymästä tai esityksestä ja controller toiminnasta. Malli tarjoaa sovellukselle tarvitsemat loogiset tiedot ja toiminnot. [18.]

Djangoa käytetään laajasti web-kehityksessä sen monipuolisten ominaisuuksien avulla. Django-ohjelmistokehitystä www-sivuillaan käyttäviä tahoja ovat muun muassa The Washington Times, the Public Broadcasting Service, Mozilla sekä sovellukset Instagram ja Pinterest. [17.]

Sovelluskehitysalustana Google Apps Engine

Google Apps Engine on sovelluskehitysalusta, joka on rakennettu pilvipalvelupohjaiseksi alustaksi web-sovellusten kehittämiseen ja ylläpitoon. Se käyttää pilvipalvelun tarjoamiseen PaaS-tekniikkaa (platform as a service), jonka avulla käyttäjä voi luoda sovelluksia, jotka hyödyntävät ohjelman omia palvelimia, tallennustilaa ja verkkoyhteyksiä. App Engine -sovelluksia on siis helppo rakentaa, hallinnoida ja skaalata. [19.]

4 Oppimisympäristöt

4.1 Avoin verkkokurssimuoto MOOC

Mooc eli Massive open online course tarkoittaa kaikille avointa verkosta löytyvää avoimesta materiaalista koostettua kurssia, jolle kuka tahansa voi osallistua ajasta ja

paikasta riippumatta. Kurssi on skaalautuva, eli Mooc-toteutuksilla osallistujia voi olla suuret määrät, mistä nimikin juontuu. Erisisältöisiä ja erilaisia oppimistapoja yhdistäviä Mooceja löytyy verkosta useita, ja niiden suosio on jatkuvasti kasvussa. [20.]

Yksi ensimmäisistä Mooc-kurssin kehittäjistä on Shimon Shoken, joka tarjosi kollegansa kanssa tietokoneaiheisen koostamansa kurssin materiaalin vapaasti käytettäväksi verkon välityksellä. Shokenin näkökulma tälle ensimmäiselle Mooc-kurssille oli, että opettajien lähtökohtainen tarkoitus ei ole vain luennoida ja opettaa vaan pikemminkin tarjota sopiva ympäristö ja hyvät työkalut houkutellakseen esiin jokaisen opiskelijan sisäisen oppimisen taidon. Moocien tarkoitus onkin opiskelijoiden oman aktiivisuuden kautta hyödyntää oppimisen työkaluja ja samalla tarjota tukea yhteisöjen kautta. [21.]

Moocien suursuosio alkoi vuonna 2011 syksyllä, kun Standford Univeristyn ensimmäiselle avoimelle verkkokurssille aiheesta ”tekoäly” ilmoittautui ja osallistui yli 160 000 opiskelijaa ympäri maailman [3].

Vuosi 2012 oli Moocien vuosi. Trendi lähti suuressa mittassa liikkeelle ja keräsi jopa miljoonia käyttäjiä, kun suuret yliopistot kuten Standford ja MIT halusivat järjestää Mooc-kursseja ja kehittää opetustarjontaansa myös verkossa. Nykyään useat huippuyliopistot tarjoavat videoituja huippuluennoitsijoiden luentoja, joita käyttävät johtavien online Mooc-kurssien tarjoajat, kuten edX ja Coursera. Yliopistot työskentelevät tiiviissä yhteistyössä näiden kurssitarjoajien kanssa ja ovat mukana kehityksessä. [22; 3.]

Yksi suurimmista Mooc-kurssien tarjoajista on tammikuussa 2012 perustettu Coursera. Yrityksen perustana on tarjota parhaat kurssit parhaasta opetusmateriaalista ilmaiseksi ja vapaasti kaikkien käytettäväksi. Kurssit ovat saavuttaneet valtavan suosion perustamisensa jälkeen vuoden 2012 alusta lähtien. Erinäisille kursseille ilmoittautuneita opiskelijoita on jopa 1,7 miljoonaa. Käyttäjämäärän kasvu on nopeampaa kuin sosiaalisen median edelläkävijällä Facebookilla, ja osallistujien määrä on edelleen jatkuvasti kasvussa. [20; 22.]

Courseran ajatuksena on tarjota kurssin teoriaosuudet videoluentojen muodossa, mutta passiivisten videoluentojen käyttö ei yksin riitä aktivoimaan opiskelijoita. Luentojen seuraksi ja niiden lomassa tarjotaan aiheeseen liittyviä tehtäviä, joiden avulla

oppiminen tapahtuu luonnollisesti ja kurssin sisäistäminen on helpompaa. Courseran perustajan Daphne Kollerin mukaan tutkimukset ovat osoittaneet, että harjoittelu kehittää oppimista ja jo opittujen asioiden toistaminen esimerkiksi tehtävien muodossa edistää oppimista. Muutkin Mooc-opistot ovat ymmärtäneet teorian ja sen kertaukseen laadittujen tehtävien hyödyllisyyden. [11.]

Esimerkiksi P2P-verkkoyliopisto käyttää Harvard- ja MIT-yliopistojen videoluentomateriaaleja ja yhdistää niihin Codecademyn tehtäviä. Tällaisen eri materiaalien yhdistelyn kautta voidaan tuottaa hyviä oppimisentuloksia. [12.]

Moocit toimivat sekä yliopistojen omien opiskelijoden että ulkopuolisten käyttäjien opetustyökaluna. Moocit on koostettu niin, että kurssi muodostaa yhden opintokokonaisuuden, josta on mahdollista ansaita todistus ja oppia aiheen keskeiset asiat ilman luokkahuoneessa läsnäoloa.

Suomessakin on toteutuettu Mooc-oppimista. Ensimmäinen Helsingin yliopiston tietojenkäsittelytieteen laitoksen Mooc-toteutus syntyi vuonna 2012, kun tehostetun kisällioppimisen menetelmällä toteutetun ohjelmointikurssin työskentelytapa synnytti opetustyylin, joka voitiin siirtää myös verkko-opetukseksi laajemman käyttäjäkunnan saataville. Kurssille ilmoittautui mukaan vähän yli 400 opiskelijaa. [24.]

Moocit tarjoavat tilaisuuden ylipäätään kehittää opiskelua monilla tieteenaloilla, mutta erityisesti ohjelmoinnin opiskeluun tämänkaltainen toteutus soveltuu erittäin hyvin. Moocit tarjoavat materiaalia kehittää opetusta, ja niiden tulokset antavat tietoa verkko-opiskelun kannattavuudesta. Insinöörityössä otetaan mallia teoriaosuuksien koostamiseen Mooc-kokonaisuuksista.

4.2 Koulutusteknologiat

Koulutusteknologia on jatkuvasti kasvussa, ja sille löytyy myös paljon kysyntää. Uusia yrityksiä syntyy jatkuvasti, ja niiden tarkoitus on tarjota opetusta joko ilmaiseksi rahoittamalla toimintansa mainostuloilla tai säätiöiden tuella tai keräämällä käyttäjiltä

maksu opetuksesta. Oppimisesta ja sen kehittämisestä on kiinnostuttu uudella tavalla, ja opetuksen tarjoajia on paljon. [23.]

Mooc-periaatteella koostettuja avoimia verkkokurssien tarjoajia on useita. Niistä tunnetuimpia ovat muun muassa Coursera, edX ja Udacity, jotka kaikki toimivat vahvasti yhteistyössä eri yliopistojen kanssa muodostaen linkin verkko-opiskelijoiden ja yliopisto-opiskelun välille ja tarjoten mahdollisuuden päästä osalliseksi huippuyliopistojen opetustarjonnasta. Useat yliopistot ovat liittyneet Moocien tarjoamaan opetusmuotoon ja tarjoavat opiskelijoilleen opetusta näiden yritysten kautta. Kurssit etenevät kuitenkin tietyn aikataulun mukaan myös itsenäisille verkko-opiskelijoille, ja osallistujilta odotetaan tiettyä etenemistahtia, jotta kurssin saa suoritetuksi. [23.]

Samankaltainen, mutta opiskelutahdiltaan vapaampi oppimistyökalu, jonka opetussisällöt koostuvat niin tutoriaaleista kuin myös tehtävistä, on muun muassa Khan Academy. Tällä oppimisalustalla aihetarjonta on hyvin laaja ja kurssisisältöjä on lukuisista eri aihepiireistä. Yrityksen on perustanut jo aikaisemmin mainittu, käänteisen luokkahuoneen mallia soveltanut Salman Khan. Sivusto on ilmainen kaikille käyttäjille, eikä se kerää osallistumismaksuja. [8.]

KhanAcademyn innoittamana monet yksityishenkilöt ovat päätyneet perustamaan omia startup-yrityksiä huomatessaan yleisön olevan kiinnostut heistä opetusmateriaalin vdeotutoriaalien tarjoajana. Materiaalin tuottaminen on usein alkanut innostuneesta ja harrastuksien parista asiaan kohtaan. Koulutuskulttuuri on uuden aikakauden edessä siksi, että eritaustaiset ihmiset pystyvät kehittämään taitojaan verkon välityksellä ja vapaamman viestintäverkoston avulla tietoa tuottavat virallisten tahojen lisäksi myös yksittäiset henkilöt. [8.]

Esimerkiksi videopalvelu Youtubessa erilaisia videotutoriaaleja erinäisistä aiheista on jo tuhansia, ja niitä katsotaan ahkerasti. Oppiminen videotutoriaalien kautta on usein tehokkaampaa, kun tietoa omaksutaan paremmin, kun voi käyttää sekä kuulo- että näköaistejaan verrattuna vain tekstisisältöön.

Maksullisia sivustoja opiskeluun on myös useita. Yksi niistä on ohjelmoinnin ja web-kehityksen opetukseen keskittynyt Treehouse. Myös maksullista ohjelmoinnin opetusta tarjoaa GoToLeaf, joka on keskittynyt eritysesti Ruby on Rails -ohjelmointiin.

4.3 Yhteisöllisiä työkaluja

Yhteisöllisiä työkaluja erilaisen materiaalin jakoon ja yhteistyöskentelyyn on paljon. Suurin tarjonta osa on pääpiirteisesti suunniteltu yritysten käyttöön, videochatien ja dokumenttien jakoon ja yhteisölliseen työskentelyyn. Myös koulutustahoja palvelevia työkaluja on olemassa useita.

Opetusmateriaaliin ja tässä toteutuksessa videomateriaalin jakelun tarkoitukseen tehtyjä sivustoja tai jopa koko kurssin koostamista varten tarkoitettuja sivustoja on useita. Googlen julkaisema course-builder tarjoaa hyvän alustan kurssin koostamista varten. Muita materiaalin koostamista varten tarkoitettuja kurssipohjia tarjoavat muun muassa Kune ja Coursesites, jotka tarjoavat hyvät yhteisölliset työkalut verkkoyhteisötyöskentelyyn. [25; 26; 27.]

Reaaliaikaisia selaimessa toimivia yhteisötyökaluja, joiden avulla voi toteuttaa yhteisöllistä oppimista lähiopetuksessa, on monia. Eniten työkaluja on suunniteltu jakamaan yhdessä muokattavia ajatuskarttoja tai dokumentteja, mutta myös vapaamman sisällön tuottamiseen tarkoitettuja visuaalisia työkaluja löytyy, ja niitä myös tällä kurssilla voi käyttää apuna. Maksullinen, mutta hyvät yhteisötyökalut tarjoava ratkaisu on conceptboard.com, jonka avulla voi jakaa materiaalia ja tehdä yhteisöllisesti merkintöjä. Twiddla.com on myös yhteisötyökalu, jolla jakaa erisisältöistä materiaalia, mutta graafiselta käyttöliittymältään se on melko vanhanaikainen eikä toiminnoiltaan kovin laaja. Yliopistokäytössäkin testattu scribblar.com-työkalu on toteutettu yhteisölliseen työskentelyyn, mutta painottuu yleisilmeeltään enemmänkin chat-työtilana kuin visualisena yhteisötyökaluna. Vielä beta vaiheessa oleva stixy.comin työkalu on kätevä ratkaisu kuvien ja muistiinpanojen jakoa varten. [28; 29; 30; 31.]

Ohjelmoinnin oppimista edistävä web-työkalu on Copmilr, joka sallii käyttäjän luoda ohjelmointiprojekteja verkossa selaimellaan. Yrityksen tarkoituksena on rakentaa alusta, joka kokoaa pilvipalvelupohjaisia ohjelmointityökaluja ja tarjoaa monimuotoisen ohjelmointiympäristön. Compilr-startup-yritys ja vielä tällä hetkellä ilmainen, mutta vastaisuudessa suunnitteilla on maksullinen versio. [32.]

4.4 Python-ohjelmoinnin opetus

Ohjelmoinnin harjoittelua varten on useita virtuaalisia ohjelmointialustoja. Python on muiden muassa suosittu ohjelmoinnin opiskelun aloituskieli. Harjoitusalustoja on learnpythonin.orgin sivustoilla. Niissä oppiminen on koostettu teorasta ja sivun alalaidassa interaktiivisesta ohjelmointityökalusta, jolla voi testata koodin käyttäytymistä. Opetusta on myös maksullisena versiona. Esimerkiksi learnpythonthehardway.org-kurssi koostuu videotutoriaaleista ja tehtävistä. Codingbat tarjoaa ohjelmoinnin opetusta yksinkertaisella, mutta asiasisältöisellä sivustolla. Sivusto tarjoaa paljon esimerkkejä ja johdattelua ohjelmointiin, mutta graafiselta käyttöliittymältään se ei ole edukseen. Learnstreet on vielä beta-vaiheessa oleva, mutta hyödyllistä ohjelmointiopetusta myös Pythonista tarjoava starup-yritys, joka jatkokehityksen jälkeen tarjoaa varsin varteenotettavan alustan oppimiselle. [33; 34; 35; 36.]

Ohjelmointia on melko yksinkertaista opettaa tutoriaalien välityksellä, sillä aiheet ovat hyvin teoriapohjaisia. Siksi videotutoriaaleja ohjelmoinnin opetuksesta löytyy verkosta melko laajasti erilaisia ja eritasoisilta tarjoajilta.

LeanersTv sisältää muiden aiheiden muassa ilmaisia videotutoriaaleja, ja Python-videot ovat ThenewBoston Educationin tuottamia. Showmedo on vapaan lähdekoodin yhteisö, joka tarjoaa tuottamiaan Python-videoita käyttäjille. Sivustolla on laaja Python-sarja, mutta käyttöliittymältään ja videontoistoltaan se on melko heikko. Hyvä ja selkeä videotutoriaalisarja Pythonista on tarjolla newthinkthank.com-sivustolla, jonka tekijä Derek Banas ohjaa läpi Python-aiheiden esimerkkien avulla. Python-videotutoriaaleja on myös Youtubessa tarjolla MIT open coursewarelta, jossa aihe on opetettu oikean luentomateriaalin pohjalta, jossa luennoitsija John Guttag selittää Python-ohjelmoinnin keskeisiä aiheita liitutaulua apuna käyttäen. Samalla tyylillä, luentotallenteella Google Python class -opintototeutukselta taltiointina, Nick Parlante opettaa Python-ohjelmointia. [37; 38; 39.]

Myös Pythonin ja Djangon yhdistäviä ohjelmointikursseja on jonkin verran. Esimerkiksi Udemy tarjoaa erilaisia versioita kursseillaan Pythonista, enimmäkseen aloittelijoille. Kurssit ovat maksullisia ja tarjoavat kattavat videotutoriaalit aiheesta. Myös Nettuts+ tarjoaa opetussisällössään Pythonin ja Djangon yhdistävää opetusta. Materiaalit ovat ilmaisia yksityishenkilöille, mutta vaativat kirjautumisen sivustolle. [40.]

Python-aiheisia videotutoriaaleja ja oppimismateriaalia on tarjolla sekä opetukseen suuntautuneilta yrityksiltä että aiheesta kiinnostuneilta yksityishenkilöiltä. Python-ohjelmointikielen opetusta on tarjolla sekä versiosta Python 3 kuin myös vanhemmasta Python 2:sta. Selkeästi useampi on kuitenkin panostanut Python 2:n opetukseen, koska se on vielä yleisempi ja yhteensopivampi.

5 Web-ohjelmoinnin opintojakson koostaminen

5.1 Valitut työkalut ja opetusmateriaalit

Web-ohjelmoinnin opintojakson koostamiseen valittiin oppimisympäristö, joka koostuu opintojakson materiaalin jäsentelevästä ja helposti asiakokonaisuuksia hahmottavasta Todo-sivustosta, johon kurssimateriaali on koottu. Sivusto tarjoaa yhteisöllisen työvälineen edistymisen seurantaan ja tiedon jakamiseen kommentoinnin ja chatin avulla. Sivustolle on helppo koostaa materiaalit, ja se toimii linkkinä opiskelijoille.

Videomateriaalit on koottu Google Sites -palveluun, johon on helppo navigoida Todo-kurssiympäristön kautta. Videoiden opetussisällön tueksi käytetään harjoitussivustoa Codecademy, joka sisältää laajan harjoitustehtäväsisällön Pythonin opiskeluun ja mahdollisuuden reaaliaikaiseen oman koodin tuottamisen harjoituksissa.

Tuntityöskentelyn yhteisölliseksi työkaluksi valittiin CanvasDropr. Se on työväline helpottamaan laboratoriotunteja, sillä sen avulla voi jakaa tiedostoja ja materiaalia.

Todo-kurssiympäristö

Todo-kurssiympäristö on Metropolia Ammattikorkeakoulun päätoimisen tuntiopettaja Antti Laihon suunnittelema ja kehittämä oppimisalusta. Tämä kurssiympäristö on luotu osana ylemmän ammattikorkeakoulututkinnon opinnäytetyötä.

Todo-sivusto syntyi, kun Antti Laiho, 3D-mallintamisen ja peliohjelmoinnin opettaja, alkoi pohtia ja kehittää ketterämpää ratkaisua oppimisalustaksi vetämilleen opintojaksoille. Aiemmilla opintojakson toteutuskerroilla oli käytetty Metropolian oman

kurssiportaalin Tuubin työtilaa materiaalien jakoon, tehtävienantoon ja palautustöihin. Tuubi tarjoaa luotettavan ja melko selkeän kanavan opintojakson materiaalien jakoon, mutta opintojaksoille, jotka sisältävät paljon informaatiota ja tehtäviä, Tuubi on melko kömpelö. [41.]

Lähtökohtainen kysymys Todon kehityksessä oli, kuinka voisi parantaa opetusta ja luoda opetusalustan, joka olisi ketterä opettajien työväline opintojakson tehtäviä varten ja tarjoaisi myös selkeän alustan opiskelijoille hahmottaa opintojakso kokonaisuudessaan. [41.]

Toisin kuin aikaisemmat opintojaksot, jotka koostuivat laboratoriotyöskentelystä ja opintojakson lopussa palautettavasta tehtävästä, Todo-sivustolle toteutettu pelikurssi koostuu Todo-tehtävälistasta, jossa koko opintojakson tehtävät ja materiaalit on hajautettu pienempiin osatehtäviin, jotka tähtäävät koko kurssin suorittamiseen tarvittavaan harjoitustyöhön. Opiskelija siis kasvattaa tehtävien avulla tietotasoaan ja taitojaan, joita käyttämällä voi suorittaa lopputehtävän. [41.]

Tehtävät on koottu listaan lineaarisella periaatteella. Tietotason karttuessa opiskelija voi siirtyä haastavampiin tehtäviin ja näin edetä omaan tahtiinsa. Opintojakson tehtävälistassa luetellut tehtävät ovat kaikki pakollisia, mutta jokaisella opiskelijalla on mahdollisuus suorittaa tehtävät omaan tahtiinsa. Kun tehtävä on suoritettu, opiskelija merkitsee sivustolle tehtävän tehdyksi. [41.]

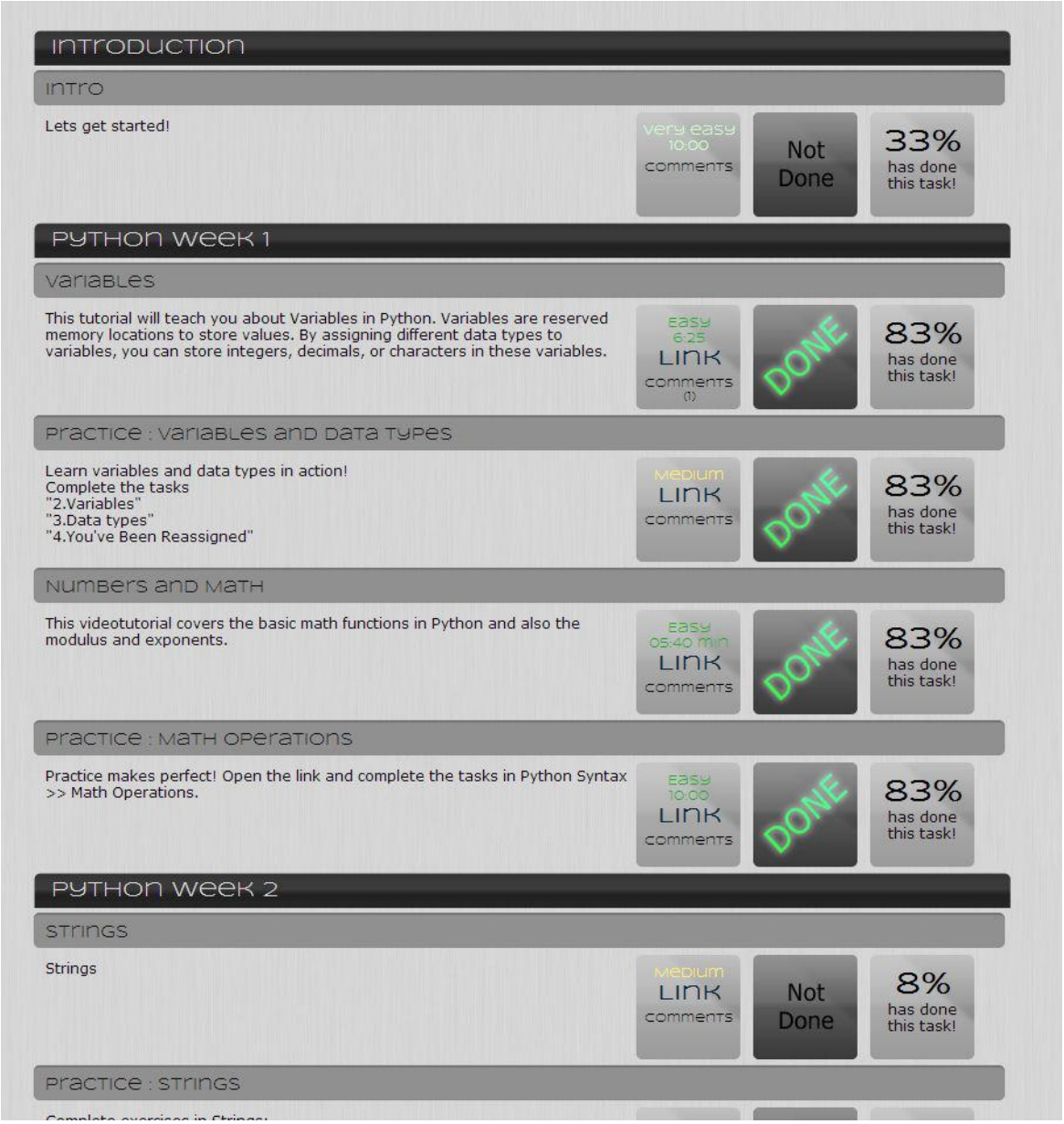
Vaikka tehtävälistaus on kaikille kurssisivulle kirjautuneille opiskelijoille yksilöllinen, on sivustolla mahdollisuus myös vuorovaikutukseen muiden kurssille osallistujien kanssa. Jokaisella kurssitoteutuksella on oma chat-työtilansa, jonka avulla näkee muut online-tilassa olevat opiskelijat. Sivusto tarjoaa siis mahdollisuuden keskusteluun ja vuorovaikutukseen muiden opiskelijoiden kanssa ja mahdollisuuden saada vastauksia esimerkiksi tehtäviin liittyviin kysymyksiin ja ongelmakohtiin. Näin myös opiskelijat voivat tarvittaessa toimia tutoreina toisilleen. Sivuston chat on myös opettajalle hyvä työkalu päivystää sivustolla ja auttaa ja ohjata opiskelijoita, jotka ovat valinneet etätyöskentelymahdollisuuden. [41.]

Jokaiseen yksittäiseen tehtävään on liitetty mahdollisuus kommentoida tehtävänantoa tai kirjoittaa kysymyksiä niiden ilmetessä. Opettaja voi vastata näihin kommentteihin, ja keskustelu on näkyvillä kaikille opintojaksolle osallistujille. Kommentointi chat-

mahdollisuuden lisäksi toimii siis tiedon välittäjänä kaikille, joita sama kysymys askarruttaa. [41.]

Opintojaksolle osallistujien välille luo yhteisöllisyyttä toiminto, jolla voi seurata muiden opintojaksolle osallistujien edistymistä. Kun opiskelija on saanut tietyn tehtävän tehdyksi ja merkitsee sen järjestelmään klikaten nappia ”done”, järjestelmä laskee prosentuaalisen määrän niistä kurssilaisista, jotka ovat tehtävän jo suorittaneet. Näin opiskelijat voivat seurata myös muiden edistymistahtia ja saada käsitystä edistymisestään muihin verrattuna. Tämä yhteisöllisyys ja tietynlainen ryhmäpaine kannustaa opiskelijoita edistymään tehtävissään, ja mielikuva yhteisöllisestä tekemisetä auttaa etenemisessä, vaikka opintojakso suorittettaisiinkin täysin itseopiskeluna. [41.]

Kuvassa 1 on näkymä Todo-sivuston käyttöliittymästä opiskelijan näkökulmasta. Pääaiheet on merkitty mustalla taustalla, ja niiden alle koostuvat viikkokokonaisuuteen kuuluvat tehtävät. Erilliset tehtävät koostuvat linkityksestä videomateriaaliin, ja jokaisen videomateriaalin alla on linkki harjoitustehtäväsivustolle. Kuvassa on näkymä, jossa opiskelija on suorittanut Python week1:n alle koostetut neljä tehtävää ja merkinnyt ne tehdyiksi. Oikealla puolella näkyy, että myös 83 prosenttia muista opintojaksolle osallistujista on jo ehtinyt suorittaa samat tehtäväannot.



Kuva 1: Todo-sivuston käyttöliittymä.

Kurssisivusto on toteutettu PHP:n jQueryn ja MySQL:n avulla. Suunnittelussa ja koostamisessa on noudatettu periaatetta toteuttaa sivusto mahdollisimman yksinkertaisesti, jotta lopputulos olisi selkeä niin opiskelijan kuin opettajan näkökulmasta. [41.]

Sivusto on varsin uusi, sillä se on otettu käyttöön vuoden 2013 alussa. Ensimmäisiä toteutettuja opintojaksoja ovat olleet 3D-mallinnus ja pelikurssit Unity Basics ja jatkokurssi Unity Mmo Basics. Sivusto on otettu hyvin vastaan, ja opiskelijat ovat olleet tyytyväisiä opintojaksojen toteutukseen. Myös oppimisympäristö on jo saanut

positiivista palautetta opiskelijoilta muun muassa sen eduista sovittaa yhteen itseopiskelu kurssisivuston tuella.

Web-ohjelmoinnin opintojakson koostamisessa päädyttiin käyttämään Todo-sivustoa juuri sen selkeyden ja tehtävien jaottelun vuoksi. Myös yhteisöllisyys, jossa opiskelijat näkevät oman edistymisensä lisäksi muiden edistymisen tehtävien suorituksessa, koettiin tärkeäksi välineen valinnassa.

Todo-sivuston avulla tuetaan ajatusta opiskelijoiden itseohjautuvuudesta ja vapaudesta valita, missä ja milloin opintojaksoon kuuluvat osatehtävät suorittaa. Tehtävät on kuitenkin ryhmitelty viikkokokonaisuuksiksi, joiden aikarajoituksia on noudatettava. Lähiopetustunnille vaaditaan, että viikon tutoriaalit, ja tehtävät ovat suoritettu ja opettaja pystyy seuraamaan prosenttilukujen perusteella, että tehtävät on suoritettu.

Vaikka tehtävien suoritetuksi merkitseminen onkin jokaisen opiskelijan oman rehellisyyden varassa, on ominaisuus myös opiskelijoita aktivoiva. Opiskelijan rooli omassa oppimisprosessissa on enemmän itseohjautuva, ja jokainen voi ottaa vastuuta omasta edistymisestään ja oppimisprosessistaan.

Python-ohjelmoinnin videotutoriaalit

Python-ohjelmoinnin videotutoriaalien tarjoaja on TheNewBoston Education. Sivusto tarjoaa videotutoriaaleja laajalla alueella erialisista aiheista keskittyen erityisesti ohjelmoinnin opetukseen. Materiaalia on tarjolla ilmaiseksi kaikille oppimisesta kiinnostuneille, eikä vaadi minkäänlaista rekisteröitymistä tai rajoita materiaalin käyttöä. [42]

Videomateriaali välittyy käyttäjille videonjakopalvelu Youtuben kautta. Python-videotutoriaaleja on yhteensä 43, ja ne kattavat Pythonin ohjelmointiaiheet perusteista monimutkaisempiin ohjelmointiaiheisiin. Tutoriaalit perustuvat reaaliaikaiseen ohjelmointiin. Tutoriaaleissa ohjaaja käyttää ohjelmointiin Python idle – ohjelmointyökalua, joka toteuttaa koodirivin heti käyttäjän sen syötettyä. Videon näkymää vallitsee ohjelmointiympäristö ja opettaja on itse taka-alalla. Videon opetusmateriaali on koostettu Python-versiosta 2.7. [42.]

Opetuksessa käytetään paljon havainnollistavia esimerkkejä, ja ohjelmoinnin teoria esitetään käytännön kautta. Python-ohjelmoinnin aihealueet on jaettu tasaisesti noin 3—7 minuutin pituisiin aihekokonaisuuksiin, jotka etenevät loogisesti rakentaen osaamista aiemmin opitun varaan. Tutoriaali on muita Python-materiaaleja vähemmän muodollinen, ja se tekee videoista helposti lähestyttävän. Tutoriaaleja on miellyttävä kuunnella kun opetus ottaa opiskelijan selkeästi huomioon ja on juuri opiskelijalle suunnattua eikä luennointityylistä. Puhetyyliltään video on rentoa, ja aiheet selitetään auki helposti ymmärrettävästi selkein termein.

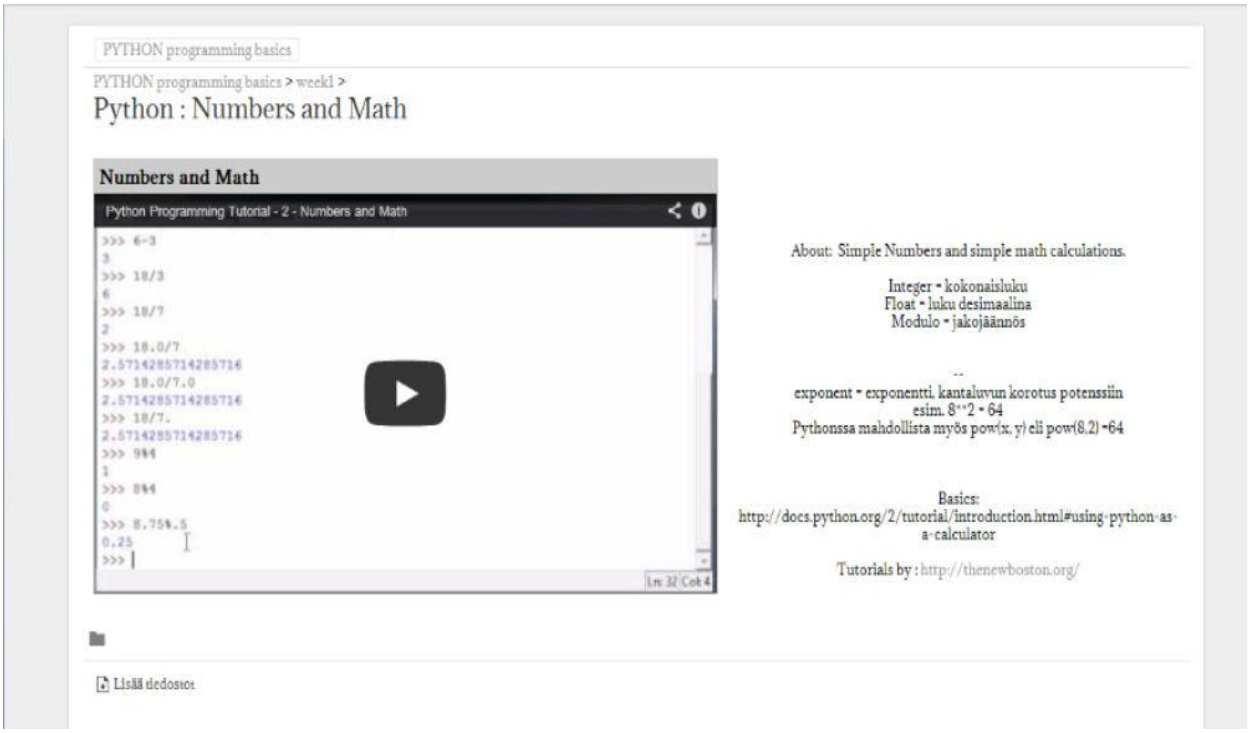
Koska tutoriaalit lähtevät liikkeelle yksinkertaisimmista perusteista, ei niiden seuraamiseksi ja ymmärtämiseksi tarvita minkäänlaista aikaisempaa ohjelmointikokemusta Pythonista. Asiat selitetään yksinkertaisesti alusta alkaen, ja ohjelmointiin perehtyvän opiskelijan on helppo seurata.

Videojakelu Google Sites -palvelun avulla

Todo-sivusto on tarkoitettu muun muassa erilaisten tehtävien kokoamiseen ja ohjeistukseen, ja sen avulla opiskelija voi suorittaa asiat itsenäisesti. Sivuston kautta voi linkittää materiaalia suoraan tehtävänantoon, mutta erillisen materiaalin, kuten videoiden ja tekstitiedostojen, upottaminen ei ole mahdollista ainakaan vielä tässä vaiheessa kehitystä.

Videoiden selkeään viikkokokonaisuuksien jaotteluun ja jakeluun opiskelijoille otettiin käyttöön Google Sites -palvelu, johon videot voitiin upottaa suoraan Youtubesta ja tarvittaessa lisätä lisäohjeistusta, kuten ohjelmointisanastoa tekstin muodossa. Sites tarjoaa myös suoraan Youtubesta katsottuun videoon verrattuna paremman mahdollisuuden opiskelijalle löytää oikea materiaali ja keskittyä ilman sivupolkuja.

Kuvassa 2 on näkymä Google Sitesiin koostetusta materiaalista. Kuvassa on upotettu videomateriaali ja oikealla puolella sanastoa kertomaan videon sisällöstä ja linkkejä aiheen syventämiseen.

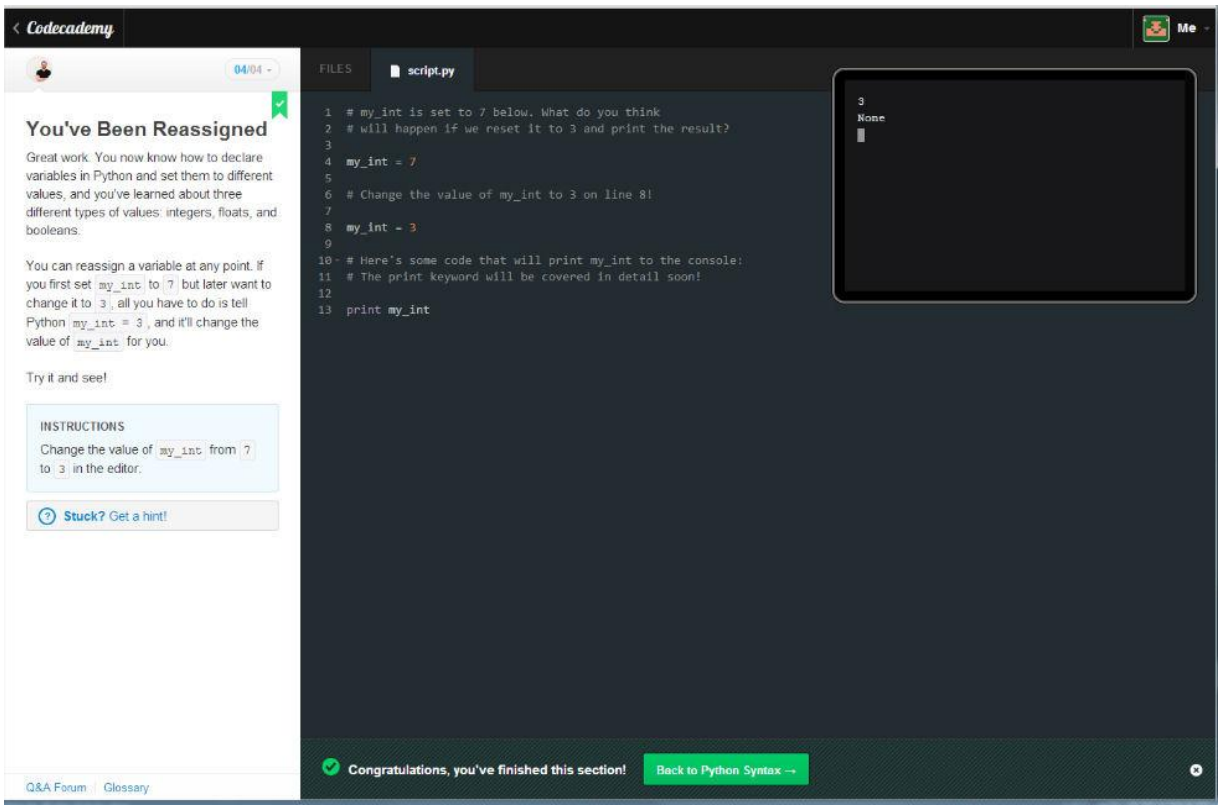


Kuva 2: Google Site-sivusto, jolle on upotettu Youtuben videotutoriaali.

Interaktiivinen ohjelmointialusta Codecademy

Codecademy on interaktiivinen ohjelmointialusta, jonka tavoite on opettaa ohjelmointia eritasoisille oppijoille. Codecademyn ovat perustaneet Zach Sims ja Ryan Bubinski vuonna 2011. Se sisältää ohjelmointikursseja monesta eri kielestä, kuten Python, JavaScript ja Ruby. Sivusto opettaa onnistuneesti ohjelmointia harjoitusten avulla, ja menestyksestä kertoo se, että kursseilla on yli 550 000 osallistujaa. Sivustolla voi suorittaa kokonaisia kursseja eri ohjelmointikielistä ja seurata omaa edistymistään keräämällä suoritusmerkkejä suoritetuista tehtäväkokonaisuuksista. [43.]

Kuvassa 3 on näkymä Codecademyn graafisesta käyttöliittymästä tehtävän suorituksen aikana. Vasemmalla näkyy tehtävän pohjustuksena ohjelmoinnin teoriaa kirjallisesti ja sen alla ohjeet tehtävän suorittamiseen. Keskellä sinisellä pohjalla näkyy valmis pohja, jota opiskelija tehtävänannon mukaan muokkaa. Tehtävässä harjoitellaan ohjelmointia muuttujilla ja muuttujien arvojen muokkaamista. Tehtävä on tehty ohjeistuksen mukaisesti, ja oikealla yläkulmassa näkyy neliön sisässä vastaus, joka on saatu aikaan ja tehtävä on suoritettu.



Kuva 3: Codecademyn ohjelmointiympäristönäkymä.

Codecademyssä on paljon tehtäviä Python-ohjelmointikielestä, ja ne on jaoteltu eriaiheisten otsikoiden alle. Web-ohjelmoinnin opintojaksolle tehtäviä valittiin Pythonin eri osa-alueista sen mukaan, miten ne soveltuvat tämän opintojakson sisältöön. Sivustolla suoritettavat tehtävät ovat hyvin havainnollisia ja kannustavat harjoittelemaan ohjelmointia. Opiskelija voi halutessaan harjoittaa ohjelmointitaitojaan lisää.

Monet ohjelmointityökalut sisältävät myös hyödyllisiä ohjelmointitehtäviä ja esittävät teoriaa ymmärrettävästi, mutta käyttöliittymältään ne eivät kuitenkaan toimi yhtä selkeästi kuin Codecademy.

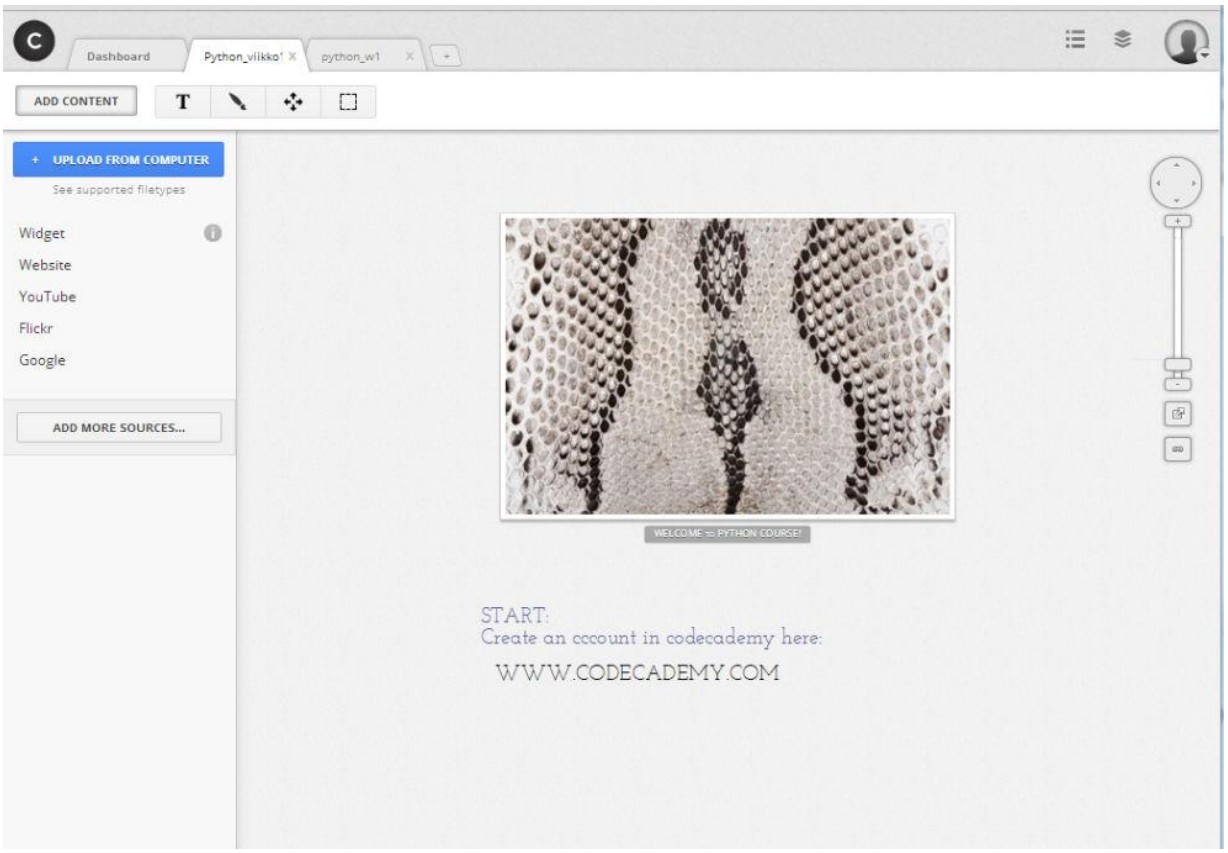
Virtuaalinen kokouspöytä Canvasdropr

Canvasdropr on web-pohjainen yhteisötyökalu. Se on kuin virtuaalinen kokouspöytä, jonka avulla voi jakaa materiaalia, kirjoittaa kommentteja ja linkittää sivustoja reaaliaikaisesti osallistujien kesken. Toisin kuin useimmat yhteisötyökalut, jotka

perustuvat tekstin jakamiseen, Canvasdropr on ketterämpi. Sitä on kuvailtu sekoitukseksi Dropboxia ja Google Docsia. [44.]

Canvasdropr on tanskalaisten Christian Rasmussenin ja Cathrene Andersenin vuonna 2011 kehittämä työkalu, joka on tarkoitettu yritysten, koulujen ja yksityiseen käyttöön. Yhä enenevissä määrin Canvasdropin käyttäjämäärät siirtyvät opetustarkoitukselliseen käyttöön. Canvasdropria käytetään paljon niin opiskelijoiden kuin opettajienkin kesken. [44.]

Kuvassa 4 on Canvasdroprin näkymä koostettuna valmiiksi ensimmäistä opetuskokeilua varten. Canvasdroprin sisältö vastaa suurin piirtein ensimmäiselle viikkokokonaisuudelle koostettavaa Canvasdropria.



Kuva 4: Canvasdropin näkymä käyttäjälle.

Canvasdropr valittiin kurssityökaluksi, koska se on helppokäyttöinen ja joustava. Sen tarkoituksena on myös säilyttää jokaisen laboratoriotunnin tiedot ja vastaukset, joihin on helppo palata ja hakea tietoa aikaisempien kertojen materiaalista.

Canvasdropr toimii tuntityöskentelyssä apuna, koska se mahdollistaa linkkien välityksen ja sen vahvuuksia on hyvin visuaalinen käyttöliittymä ja mahdollisuus jakaa materiaalia yhteisesti käytetyn alustan eli canvaksen avulla. [45.]

5.2 Tekninen toteutus

Web-ohjelmoinnin opintojakson koostamisessa lähdettiin liikkeelle videotutoriaalin jäsentelemistä aihe-alueista. Alueet jaettiin ensin sopiviin viikkokokonaisuuksiin aiheiden sivuavuuden ja sopivien työmäärien mukaisesti. Jokaiselle opiskeluviikolle suunniteltiin suunnilleen toisiaan vastaavat kokonaisuudet ja niihin liittyvät tehtävät, jotta aihealueet pysyvät tasapainoisina keskenään.

Videomateriaalit ja Codecademyn tehtävät eivät ole yhteydessä toisiinsa, eivätkä ne etene toisiinsa nähden aihealueissaan samanaikaisesti tai lineaarisesti. Tehtävät on siis linkitetty myötäilemään videomateriaalin sisältöä. Tarkoituksena on saada opiskelijoiden videoista opiskelemat aiheet toistumaan hieman eri muodossa tehtävien parissa, jotta toiston myötä opittu asia jää paremmin muistiin ja opiskelija saa myös käyttää soveltavia taitojaan.

Laboratoriotehtävät toteutetaan viikkokokonaisuuksien aiheiden pohjalta, ja niissä kerrataan myös aiheita, joita ei Codecademyn tehtävissä välttämättä ole vielä käsitelty. Koska kaikkia osa-alueita, joita videoissa käydään läpi, ei löydy harjoituksista, palataan niihin vielä laboratoriotuntien sisällössä. Periaatteena kuitenkin on, että kaikki viikkokokonaisuuksien uudet asiat esitellään ensin teoriavideoilla, ja jälkeenpäin ne harjoitellaan tehtävien avulla. Codecademyn tehtävissä ei kuitenkaan esitellä tai yritetä opettaa opiskelijoille uusia asioita, vaan työkalua käytetään vain opitun kertaamiseen.

Opintojakson ensimmäiset viikot muodostuvat helpommista, perustason Python-ohjelmoinnin aiheista, ja opintojakson loppupuoli painottuu entistä haastavimpiin ohjelmointisovelluksiin. Viikkosuunnitelma on jäsennelty taulukossa 1. Ensimmäisillä viikoilla käydään opetusohjelmassa läpi ohjelmoinnin perusrakenteita ja funktioita.

Opetus etenee rakentaen osaamista viikoittain uusiin aiheisiin, jotka opitaan edellisten aihealueiden avulla. Puolessa välissä aiheita siirrytään funktionaalisempaan ohjelmointiin ja olio-orientoituneeseen ohjelmointiin ja sen avulla rakennettaviin yksinkertaisiin sovelluksiin.

Taulukko 1. Web-ohjelmoinnin opintojakson viikkosuunnitelma.

Viikko-ohjelma	**Selitys**
1: Variables & Numbers and Math	muuttujat ja laskuoperaatiot
2: Strings & Raw_input	merkkijonot ja tekstikentät
3: Sequenses and Lists & Editing and Slicing	listat ja niiden editointi
4:Lists & List Functions	listat ja niiden funktiot
5: Methods & Tuples	metodit
6: String Methods & Dictionary	metodit ja sanakirjat
7: If else	if else -toiminnot
8: Comparisons & And and Or	vertailuoperaatiot
9: For and While Loops & Building functions	silmukat ja niien käyttö
10 Parameters	parametrit
11 Object Oriented Programming	olio ohjelmointi
12 Classes	luokat ja alaluokat
13 Constructors	rakenteellisuudet
14 Modules	moduulit

Opintojakson alkupuolella esitellään myös videotutoriaalin ja tuntityöskentelyn johdattelemana ohjelmistokehitysympäristö Django, jota käytetään laboratoriotyöskentelyssä. Djangon avulla tunneilla päästään kehittämään laajempia oliopohjaisia ja julkaisukelpoisia web-sovelluksia verkkosivustoille

Laboratoriotuntien ohjemointityökaluna käytetään Google App Engineä. Sillä on helppo rakentaa, hallinnoida ja jäsennellä koodia.

Viikkokokonaisuuksien muodostamisen perusteella rakennettiin web-ohjelmoinnin perusteiden opintojaksokokonaisuus Todo-sivustolle. Todo-sivustolla admin-tason käyttäjä pystyy hallinnoimaan kurssisisältöä ja linkittämään materiaalia tehtäviin.

Opintojakson sisällöt koostetaan yksittäisten tehtävien luonnista, jotka koostetaan eritasoisten yläotsikoiden alle. Yläotsikot voivat koostua erilaisista aihekokonaisuuksista, ja web-ohjelmoinnin perusteet - opintojakson sisältö koostettiin viikkonumeroinnin alle, josta on helppo viikonnumeroinnin mukaan seurata tarvittavien tehtävien listausta.

Yksittäiselle tehtävälle määritellään tehtävän aihe ja se, minkä yläotsikon alle se kuuluu. Tehtävään kirjoitetaan myös kuvaus, joka kertoo tehtävän sisällöstä ja lisätietoa aiheesta. Tehtäviin liitetään myös linkitys, jonka avulla opiskelija löytää oikean materiaalin. Tehtäviin määritellään myös aika, joka on opintojakson koostajan arvio tehtävään kuluvasta ajasta, ja vaikeustaso.

Todo-listan yksittäisten videotutoriaalitehtävien linkityksestä siirrytään Google Sites web-ohjelmoinnin perusteet -sivustolle, johon video on upotettu. Video on upotettu iFrame Gadgetin avulla Youtuben tarjoamalla embed-linkillä. Resoluutioksi on valittu 640 x 360 pikseliä. Tietyissä videoissa on myös käytetty aikajanamerkintää, jonka avulla video saadaan alkamaan tietystä kohdasta määriteltynä aikajanan mukaan. Näin video alkaa oikeasta kohdasta oleellisen asian mukaan. Jokaiselle yksittäiselle videolle on luotu oma alasivu, ja alasivut on määritelty viikkonumeroinnin mukaan.

Sivustolle, jolle videotutoriaalit on upotettu, on varattu myös tila lisäkommenteille heti videon läheisyyteen. Tekstikenttä tuo mahdollisuuden helpottaa tutoriaalivideon

materiaalin ymmärtämistä tarjoamalla esimerkiksi sanastoapua ja sisällön teorian muistiinpanoja. Kun tarjotaan myös linkkejä syventäviin aiheisiin, opiskelija voi perehtyä opetettavaan asiaan paremmin videotutoriaalin ohella.

Sivustolle tehtiin ulkoasultaan mahdollisimman yksinkertainen ja selkeä graafinen ilme, jolla korostetaan videota ja sen sanastotekstikenttää. Sivustolle pääsee suoraan Todo-listan linkistä, ja navigointi on helpointa juuri Todo-sivuston kautta.

Lineaarisesti edettässä Todo-sivustolla jokaisen videotutoriaalitehtävän jälkeen on tehtävälistalla harjoitustehtävämateriaalia aiheeseen. Harjoitustehtävät on erotettu videomateriaalitehtävistä otsikossa esiintyvän ”Practice:”-alkuliitteen avulla. Harjoitustehtävän linkki vie Codecademy-sivuston tiettyyn tehtävään tai tehtäväkokonaisuuteen, joka opiskelijan on suoritettava. Tehtäviä saattaa olla useitakin yhteen aiheeseen liittyen, ja ne on sijoiteltu Todo-sivustolle eri linkkien alle, jotta opiskelija löytää suoraan tarkoitettuun aihealueeseen.

Toisin kuin aiemmin Todo-sivustolle koostetut opintojaksot, joiden suorittaminen sallii tehtävien suorittamisen jokaisen opiskelijan omaan tahtiin, tämä opintokakso on toteutettu etenemään viikkoaiheiden mukaisesti. Tämä edesauttaa vuorovaikutusta opettajan ja opiskelijoidenkin välillä, sillä kysymysten ilmetessä ovat muut samassa aiheessa ja vastauksia kysymyksiin saa opiskelijakollegoilta. Viikoittaisen yhteisen etenemistahdin hyvä puoli näkyy myös chatin käytössä, sillä opiskelijat ovat aina keskenään samassa aiheessa ja ohjeita on helppo kysellä kanssaopiskelijoilta.

Canvasdropin toimintoja käytetään laboratoriotunneilla. Materiaali koostuu tunnin sisällön mukaan. Opettaja luo jokaista tuntia varten uuden Canvaksen, johon tunnin aikana lisätään tehtävänantoja ja kysymyksiä. Opetuskokeilua varten Canvasdropriin luotiin viikon 1 materiaaleja vastaava Canvas-sivu, jolle lisättiin tunnilla tarvittavat linkit ja kuvat. Canvasdropria on mahdollisuus käyttää suljettuna työtilana, johon sähköpostin välityksellä lähetetyllä kutsulla voi päästä osallistumaan. Opintojaksoa varten koostettiin avoimen linkin Canvas, jolle kaikki halukkaat pääsevät osallistumaan oikean linkin avulla. Canvakset säilyvät sivustolla myös käytön jälkeen, ja niille voi palata tarvittaessa.

Laboratoriotehtävät koostetaan kertaamaan videomateriaalia ja Codecademy-harjoitusten sisältöä. Laboratoriotunneilla ei oteta käsittelyyn täysin uusia aiheita tai

opeteta uudelleen periaatteita, jotka on käyty jo ennakkomateriaaleissa läpi. Tehtävien pohjustuksessa voidaan palata tiettyihin keskeisiin asioihin, jotta laboratorioharjoituksissa ei tarvitsisi palata esimerkiksi syntaksiin liittyviin seikkoihin. Tavoitteena on ennemminkin asian kokonaisuuden omaksuminen ja sen pohjalta harjoitustöiden suorittaminen.

Harjoituksissa laaditaan muun muassa yksinkertaisia ohjelmointitehtäviä, jotka opiskelijat voivat suorittaa yksin tai mahdollisesti myös pareittain. Tehtävänantona voi toimia esimerkiksi harjoitus, jossa tehtävän haluttu lopputulos on annettu ja opiskelijat saavat itse toteuttaa siihen tähtäävän ratkaisun. Opettaja esittää oman versionsa tehtävästä, ja erilaisia ratkaisuja voidaan vertailla ja oppia ongelmanratkaisukykyä ja opitun soveltamista.

Koska tavoitteellinen ohjelmoinnin opetus tähtää ohjelmoinnin perusteiden hallintaan ja ohjelmoinnin toimintaperiaatteiden ymmärtämiseen sekä kykyyn soveltaa opittuja asioita, pitäisi opinojakson lopputentin testata opiskelijoiden osaamista tästä näkökulmasta. Aiemmin tentti on suoritettu koodaamalla tenttitehtävät valvotussa tilassa ulkomuistista ilman Internetin tai kurssimateriaalin apua. Voisi olla hyödyllistä kokeilla web-ohjelmoinnin opintojaksolla uutta tenttimistapaa ja antaa mahdollisuus käyttää apuna kurssimateriaalia valvotussa tilassa. Tenttitehtävä voisi siis olla hieman laajempi, ja se arvioisi enemmänkin opiskelijan tiedonhakuvalmiuksia ja soveltamiskykyä kuin mekaanista ulkomuistin varassa työskentelyä.

Tällä tenttimistyylillä saataisiin opiskelijat hyödyntämään taitojaan kokonaisvaltaisemmin ja saataisiin aikaan myös ongelmanratkaisukykyjä vaativa tilanne, jollaisia varmasti tulee vastaan myös työelämässä.

Tentin pisteitys koostuisi siitä, kuinka vähällä ohjeistuksella opiskelija selviytyy annetusta tenttitehtävästään. Vähiten neuvoja ja ohjeistusta tarvinnut ja tehtävän toimivalla tavalla suorittanut opiskelija saisi siis parhaat pisteet, ja enemmän ohjeistusta opettajalta tai kirjallisesta materiaalista tarvinnut opiskelijan kokonaispistemäärä jäisi pienemmäksi. Tavoitteena on, että jokainen opiskelija pystyy suorittamaan annetun tehtävän tietyssä aikarajassa. Tämä tyyli hyödyttää myös niitä opiskelijoita, jotka ovat panostaneet teoriamateriaalin omaksumiseen ja pystyvät helposti palauttamaan mieleen oppimateriaalin avulla keskeiset periaatteet ja tutoriaalien esimerkeissä läpikäydyt ohjelmointiratkaisut.

5.3 Testaus

Insinöörityön valmista opintjaksokokonaisuutta päästään kokeilemaan käytännössä koko opintojakson mittaisena toteutuksena ensi keväänä lukuvuoden kolmannella periodilla 2014. Opetustyökalujen ja opintojakson sisällön testaamiseksi järjestettiin kahden oppitunnin mittainen opetuskokeilu opiskelijaryhmälle, joka koostui tietotaidoiltaan eritasoisista mediatekniikan, Media Engineering- sekä Information Technology -koulutusohjelmien insinööriopiskelijoista eri vuosikursseilta. Opetuskokeilun toteutus- ja opetuskielenä oli englanti.

Opetuskokeilun tavoitteena oli testata opiskelijoiden näkökulmasta teorialuennoista luopumisen mahdollisuutta ohjelmoinnin opetuksessa. Oppitunnin aikana kokeiltiin myös opintojaksolle valittujen opetustyökalujen toimivuutta. Opiskelijoilla oli tunnin aikana mahdollisuus tutustua käänteisen luokkahuoneen mukaisesti toteutettuun opintojaksokokonaisuuden materiaaliin ja harjoitustehtäviin. Opiskelijoita pyydettiin tunnin lopuksi arvioimaan käyttökokemuksiaan ja vaikutelmiaan tuntisisällöstä.

Koska käänteisen luokkahuoneen opetusmalli ei ole suurimmalle osalle opiskelijoista entuudesta tuttu, ei opetuskokeilukerralle osallistuvalta ryhmältä vaadittu materiaaliin ennalta perehtymistä. Opetusmallin mukaiselle ennen oppituntia omaksuttavalle videotutoriaalille ja tehtävien muodossa olevalle teoriaosuudelle varattiin vaadittava aika heti oppitunnin alkuun ohjeistuksen saattelemana, jotta kaikki opiskelijat saivat saman lähtöasetelman opetuskokeilua varten.

Aluksi osallistujat tutustuivat Todo-sivustoon ja kirjautuivat Codecademyyn. Oppitunnin alussa he katsoivat läpi viikon teoriamateriaalin ja suorittivat harjoitukset. Materiaalin omaksumisen jälkeen tunti koostui laboratorioharjoituksista, joista ensimmäisen opiskelijat toteuttivat omaan tahtiin tehtävänannon mukaan ohjeiden avulla ja toinen tehtävä oli muotoiltu niin, että kaikki saivat antaa panoksensa lähettämällä vastaukset Canvasdroprin kautta.

Opetussisällöksi valittiin web-ohjelmoinnin perusteet -opintojakson ensimmäisen viikon opetusmateriaali, jonka aiheena on Python-ohjelmointikielen muuttujat ja lukuarvojen laskuoperaatiot. Aiheeksi valittiin juuri opintojakson ensimmäisen viikon aihealue,

koska osallistujilta ei vaadittu aikaisempaa kokemusta Python-ohjelmointikielestä, ja ensimmäisen viikon materiaali oli helposti omaksuttavaa ilman minkäänlaista aikaisempaa ohjelmointikokemusta.

Google App Enginen asennusta ei ehditty saamaan Leppävaaran helpdeskiltä opetuskokeilua varten, joten laaditut tuntitehtävät tehtiin Pythonin oman ohjelmointityökalun avulla, jota myös videotutoriaalin ohjelmointiesimerkeissä käytetään. Python IDLE on yksinkertainen graafinen käyttöliittymä Python-ohjelmointiin interaktiivisesti.

Opetuskerralla testattiin myös tuntityöskentelyyn tarkoitettua Canvasdropria, jonka avulla saatiin linkit ja tehtävänannot jokaiselle näkyville myös omalle koneelle. Ennen oppituntia Canvasdroprin avoimelle sivustolle koostettiin tuntityöskentelyalusta, jonka avattuaan opiskelijat saivat käyttöönsä suorat linkit rekisteröitymistä varten Todo- ja Codecademy-sivustoille.

Canvasdroprin avustuksella jokainen opiskelija sai tehtävänannon, jonka pohjalta harjoiteltiin koodin tuottamista. Tuntitehtävä koostui aihealueista, joita viikon 1 aihealueessa käytiin läpi. Tehtävä oli samaa teoriaa, mutta se vaati, että opiskelija soveltaa taitoja ja rakentaa useammasta elementistä ohjelmoinnin avulla yksinkertaisen sovelluksen, joka laski käyttäjän puhelinnumeron matemaattisen kaavan avulla. Canvasdropin avulla tehtävänanto laitettiin kaikille näkyville ja edetä sai omaan tahtiin, kuitenkin niin, että ohjaajalta sai tietyssä tahdissa apua etenemiseen. Suurin osa opiskelijoista lähti toteuttamaan tehtävää itsenäisesti, ja ohjaaja kiersi luokassa neuvoen tarvittaessa. Lopuksi Canvakselle annettiin mallivastaus.

Toinen tuntitehtävä oli lyhyempi, ja vastausten keruuseen käytettiin Cavasdropria. Opiskelijoiden oli määrä lähettää vastauksen tiettyyn laskutehtävään kaikkien nähtäväksi Canvakselle, ja vastauksia käytiin lyhyesti läpi. Tästä mallista positiivista palautetta tuli mahdollisuudesta lähettää oma vastaus anonyymisti. Osallistuminen on helpompaa, vaikkei tietäsikään välttämättä oikeaa vastausta, kun kaikki osallistuvat eikä oikea tai väärä vastaus henkilöidy kehenkään. Vastauksia voidaan käydä läpi ja keskustella koko luokan kesken.

Oppitunnin päätteeksi toteutettiin palautekysely, joka jokaisen osallistujan tuli täyttää heti yhteisten tehtävien loputtua. Palautelomakkeessa pyydettiin opiskelijoita

arvioimaan numeerisesti ja sanallisesti tunnilla käytettyjä oppimistyökaluja sekä teknisestä että oppimisen näkökulmasta. Arviointilomake on liitteenä 1. Arviointiasteikko oli 1—5 ja lisäksi oppimisen kannalta tärkeys arvioitiin.

Kyselyn perusteella opiskelijat arvioivat videotutoriaalit yhdistettynä suoritettaviin tehtäviin lähes poikkeuksetta erinomaisiksi. Erityisesti kiitosta saivat tehtävät, joiden avulla tutoriaalin asiat jäivät paremmin muistiin. Opiskelijat arvioivat myös tämän tutoriaali ja tehtävämallin oppimisen kannalta tärkeäksi osaksi.

Palautteessa kyseltiin myös opiskelijoiden mielipidettä teorialuennon korvaamisesta videotutoriaalien ja itsenäisten tehtävien avulla. Vastanneista suurin osa koki teorialuentojen korvaamisen hyvänä tai jopa parempana vaihtoehtona, kuin mitä käytäntö on aiemmin ollut. Osalla uutta mallia vieroksuvista opiskelijoista huolenaiheita herätti erityisesti se, että mahdollisuus esittää kysymyksiä kesken luennon puuttui. Teoriavideot voisivat kuitenkin toimia esimerkiksi opettajan ja opiskelijan välisellä chat-yhteydellä.

Tunnilla käytetyt videotutoriaalit ”Python Programming tutorial : variables” ja ”Python Programming tutorial: Numbers and Math” sai arvioinnissa tuloksekseen erittäin hyvä (4) ja erinomainen (5). Palautteista käy ilmi, että tutoriaalit olivat suurimmalle osalle helposti omaksuttavat, mutta arvioinnissa on myös nostettu esiin huomio äänenlaadusta ja videoluennoitsijan välttävästä artikulaatiosta, joka koettiin häiritseväksi.

Opetuskokeiluun osallistuneiden opiskelijoiden ohjelmointiosaamisen lähtötaso oli hyvin erilainen, ja osalle opiskelijoista Python oli jo entuudestaan tuttu. Vaikka tutoriaalimateriaali ja siihen liitetyt Codecademyn tehtävät saivat arvioinnissa hyvät arvosanat, kirjallisessa palautteessa kävi ilmi, että opetusmateriaali koettiin osaksi turhankin helpoksi. Myös tuntitehtävissä toivottiin mahdollisuutta tarjota edistyneimmille opiskelijoille lisätehtäviä tai haastavampaa näkökulmaa tuntitehtäviin. Myös osa opiskelijoista olisi toivonut enemmänkin harjoitustehtäviä Codecademystä.

Codecademyn sivustolla tehtäviä oli varattu yhtensä kaksi kokonaisuutta, yksi kokonaisuus kummallekin videotutoriaalille. Ohjelmointiharjoitukset valittiin vastaamaan mahdollisimman tarkasti sisältöä, jonka opiskelija oli omaksunut videoista.

Oppimisympäristö Todo sai arviointimerkintöjä erinomaisesta tyydyttävään. Erinomaisen arvioinnin antaneet opiskelijat pitivät Todo-sivustoa helppona, selkeänä ja havainnollisena. Kritiikkiä ja parannusehdotuksia sivusto sai teknisistä toiminnallisuuksista, kuten tunnusten luomisessa puuttuvista infoteksteistä ja muiden sivustolla toteutettavien opintojakson opiskelijoiden kesken jaetusta chatistä. Myös tehtävien merkitseminen suoritetuksi vain opiskelijan omaan rehellisyyteen perustuen oli epäilyn alla toimivuudessa käytännössä.

Canvasdroprin käytössä tuntityöskentelyvälineenä ilmeni haasteita opetuskokeilun aikana, mahdollisesti koska Canvas luotiin kaikille avoimeksi eivätkä toiminnallisuudet olleet tarpeeksi joustavat yhteiseen työskentelyyn. Ennen opetuskokeilua testattuna Canvasdropr toimi yhden käyttäjän muokkaamana hyvin. Kun samaa canvasta oli muokkaamassa useampi käyttäjä, ei toimivuus ollut enää ajan tasalla. Suurimmat ongelmat ilmenivät reaaliaikaisen päivityksen suhteen. Kaikki opiskelijoiden tekemät muokkaukset tai kommentit eivät tulleet näkyville, ja yhtäaikaista muokkaamista varten piti selain päivittää tasaisin väliajoin. Tämä haittasi etenkin laboratorioharjoitusten yhteistä tarkastelemista, sillä kaikkien vastaukset eivät tulleet perille.

Opiskelijoilta Canvasdropr sai palautetta arviointiskaalan molemmista päistä. Vaikka tunnilla ilmeni välillä suuriakin haasteita yhteisötyökalun käytössä, sai työkalu myös hyviä arvioita ja osa piti sitä melko tärkeänä osana opetusta. Suurin osa vastaajista oli kuitenkin sitä mieltä, että Canvasdropr tarvitsee vielä käytännössä hiomista, eikä se välttämättä ole vielä valmis tuntityöskentelyvälineeksi. Vastaajista suurin osa arvioi kuitenkin yhteisöllisen työkalun oppimisen kannalta tarpeelliseksi osaksi opetustilannetta.

Opiskelijoilta kyseltiin myös mielipidettä kotitehtävien tarpeellisuudesta vielä varmistamaan, että tunnilla opitut asiat on sisäistetty, arvioi suurin osa vastaajista kotitehtävien olevan tarpeellisia opetuksen kannalta. Mahdollisesti toimiva ratkaisu voisi myös olla tehtävien palauttaminen tunnin jälkeen opettajalle, joka varmistaisi, että läpikäydyt asiat on sisäistetty.

Opintojakson koostamisessa tulikin ilmi mahdollinen tarve tehtävien palautukseen. Tällaista toimintoa ei Todo-sivustolta löydy, sillä sivustolla koostetut opintojaksot eivät vaadi viikoittaisia ja tehtäväkohtaisia tehtävänpalautuksia. Jos tällä web-ohjelmoinnin

opintojaksolla päädytään palautettaviin kotitehtäviin, pitää palautus mahdollisesti tehdä Tuubin tai sähköpostin välityksellä.

Jos vielä ennakkotehtävien ja opintojaksolle osallistumisen lisäksi määrättäisiin kotitehtävät, olisi itsenäisiä tehtäviä mahdollisesti liikaa. Opintojakson rakenne ei pysyisi enää tasapainoisena ja opetusvauhti kärsisi.

6 Yhteenveto

Insinöörityössä tutkittiin ja kokeiltiin tieto- ja viestintäteknologian kehittymisen myötä syntyneitä uusia malleja toteuttaa oppimisympäristöjen rakentaminen ja opetuksen muotoutuminen. Tarkoituksena oli hyödyntää näitä malleja ja rakentaa ohjelmoinnin perusteiden opintojakso hyödyntäen erityisesti käänteisen luokkahuoneen ja sulautuvan oppimisen tarjoamia opetusratkaisuja, joiden avulla opetus painottuu opiskelijalähtöisemmäksi ja poispäin luennointikulttuurista.

Työ koostui mallin mukaisen opetusmateriaalien tutkimisesta ja valinnasta ja opetusalustan koostamisesta ja testauksesta. Uuden e-oppimisen mallit tarjoavat paljon mahdollisuuksia materiaalin kannalta, jota on nykyään tarjolla paljon erilaista ja erilaisilta tuottajilta.

Opetusmallin tuloksia testattiin mediatekniikan opiskelijoille järjestetyssä opetuskokeilussa, jossa tarkoitus oli saada selville opetusalustan toimivuus ja opiskelijoiden näkemyksiä uudesta opetusmallista. Opetuskokeilusta saatiin hyviä tuloksia arvioimaan opintojaksokokonaisuutta. Näitä tuloksia voidaan käyttää pohjana, kun parannetaan tulevaisuudessa järjestettävän opintojakson toimivuutta.

Opetuskokeilusta saadun palautteen perusteella voidaan käänteisen luokkahuoneen mallia soveltaa tuloksellisesti ohjelmoinnin opetuksessa. Tärkeää on kiinnittää huomiota opettajan rooliin prosessin ohjaajana, opiskelijoiden itsenäisen tiedonomaksumisen ja soveltamisen tukijana ja myös itsenäisesti omaksutun materiaalin vertaistukena virtuaalisessa ympäristössä.

Luentomallin korvaaminen videomateriaalin avulla toteutetulla oppimisella ja sopivan oppimisalustan käytöllä voidaan saada oppimisprosessi aktiiviseksi toiminnaksi ja

itseohjautuvaksi. Näin kannustetaan myös itseopiskeluun ja tarjotaan taidot uusien taitojen kehittämiseen.

Nykymuotoisessa tietoyhteiskunnassa uuden tiedon sisäistämisen ja soveltamisen taidolla on suuri painoarvo, sillä ympäristö muuttuu nopeasti tekniikan kannalta, ja uusien asioiden omaksuminen on tärkeässä asemassa opiskelijoille myös työelämän kannalta. Itsenäinen tiedon omaksuminen onkin tärkeä taito, samoin tiedonkeruu ja sen soveltaminen. Mielestäni tämän opetusmallin käyttö tarjoaa hyvät lähtökohdat aiheen opiskeluun mutta myös itseopiskelun oppimiseen ja erilaisen verkkomateriaalin käyttöön sen apuna.

Nopeisiin muutoksiin ja virikkeelliseen ympäristöön mukautuen myös ajattelumallit ovat muuttuneet aikaisemmista. Tietoa on saatavilla helposti, ja sen hakeminen on arkipäiväisempää. Oppimisen tavat muuttuvat ympäristön mukana. Sulautuvan oppimisen avulla myös opetus on mukana kehityksessä ja mukailee opiskelijan jo omaksumaa tapaa oppia asioita.

Lähteet

1 Suominen, Riitta & Nurmela, Satu. 2011. Verkko-opettaja. Helsinki: WSOYpro.

2 Manninen, Jyri. 2011 Verkko-opetuksen esihistoria. Verkkodokumentti. <http://www.sivistys.net/nakokulmat/tieteen_sana/verkko_opetuksen_esihistoria.html>Luettu 24.3.2013.

3 Olender, Ilkka. 2012. Oppimisen tulevaisuus. Verkkodokumentti. <http://sometek.fi/oppimisen-tulevaisuus-mooc-flipped-classroom-ja-muita-ilmioita/ > Luettu 4.2.2013.

4 Lätti, Mervi ja Putkuri, Päivi. 2009. Löytöretki aikuisohjauksen maailmaan. Verkkodokumentti. <http://www.karelia.fi/julkaisut/sahkoinenjulkaisu/B18_verkkojulkaisu.pdf > Luettu 16.3.2013.

5 Vainio, Leena. Blended Learning – Sulautuvaa oppimista. Verkkodokumentti <http://portal.hamk.fi/portal/page/portal/HAMK/Tutkimus_ja_kehitys/Valtakunnalliset_verkostohankkeet/Ylempi_amk_tutkinto/artikkelipankki/teemat/opetuksen_toteutus/Tab/Blended%20learning%20-%20monimuotoista%20verkko-opetuksen%20hy%C3%B6dynt%C3%A4> Luettu 25.3.2013.

6 Yksilöllisen oppimisen menetelmä. 2013. Verkkodokumentti. Matematiikan opetuksen tulevaisuus. <http://maot.fi/2013/01/yksilollisen-oppimisen-menetelma/> Luettu 10.3.2013.

7 Toivanen, Aurora. 2012. Yksilöllisen oppimisen malli Martinlaakson lukion matematiikan opetuksessa. Pro gradu –tutkielma. Helsingin Yliopisto.

8 Khan, Salman. 2011. Let's use video to reinvent education. Verkkodokumentti. TED Conferences. LLC. <http://www.ted.com/talks/salman_khan_let_s_use_video_to_reinvent_education.html> Luettu 12.1.2013.

9 Koskinen, Pekka. 2012. Jäähyväiset luennoinnille Verkkodokumentti. <http://users.jyu.fi/~pekkosk/resources/pdf/koskinen_arkhimedes_12.pdf> Luettu 20.1.2013.

10 Vihavainen, Arto, Paksula, Matti ja Luukkanen, Matti. 2011. Extreme Apprenticeship Method in Teaching Programming for Beginners. Verkkodokumentti . <http://www.cs.helsinki.fi/u/ikiistal/harjoitus/wad/p93-vihavainen.pdf> Luettu 20.1.2013.

11 What we’re learning from online education. 2012. Verkkodokumentti. TED conference, LLC.
<http://www.ted.com/talks/daphne_koller_what_we_re_learning_from_online_education.html>. Luettu 24.3.2013.

12 Metropolia Ammattikorkeakoulu mukaan EU:n Smart Campus –hankkeeseen. 2012. Verkkodokumentti. Metropolia Ammattikorkeakoulu.
<http://www.metropolia.fi/ajankohtaista/uutiset/arkisto/?tx_ttnews%5Btt_news%5D=3593&cHash=d5248953f179ae1a21c068f7e2593be6>. Luettu 2.4.2013.

13 Python. 2012. Verkkodokumetti. Python Software Foundation.
<http://python.org/about/>. Luettu 8.1.2013.

14 Measuring Software Code Quality. Verkkodokumentti. TIOBE
<http://www.tiobe.com/index.php/content/company/Home.html>. Luettu 8.4.2013.

15 Python/Syntaksi. 2011. Verkkodokumentti. Wikikirjasto.
<http://fi.wikibooks.org/wiki/Python/Syntaksi> Luettu 10.3.2013.

16 Pohjolainen, Jussi. 2008. Johdatus olio-ohjelmointiin. Verkkodokumentti.
<http://www.slideshare.net/pohjus/johdatus-olioohjelmointiin-presentation>
Luettu 10.3.2013.

17 Meet Django. 2013. Verkkodokumentti. Django Software Foundation.
<https://www.djangoproject.com/>. Luettu 5.1.2013.

19 Wesley, Chun. 2012. What is (isn't) Google App Engine? Verkkodokumentti.
<https://developers.google.com/appengine/training/intro/whatisgae>. Luettu 5.4.2013.

20 Meriläinen-Tenhu, Minna. 2013. Sano MOOC kun haluat opetusta online.
<http://www.helsinki.fi/ajankohtaista/uutisarkisto/2-2013/19-14-04-33> .
Luettu 15.2.2013.

21 Shimon Schoken: The self-organizing computer course. 2012. Verkkodokumentti. TEDtalksDirector. < http://www.youtube.com/watch?v=iE7YRHxwoDs>. Luettu 24.3.2013.

22 Pappano, Laura. 2012. The Year of the MOOC. Verkkodokumentti. The New York Times. <http://www.nytimes.com/2012/11/04/education/edlife/massive-open-online-courses-are-multiplying-at-a-rapid-pace.html?pagewanted=all&_r=0>. Luettu 1.2.2013.

23 The Best Mooc Provider. 2013. SkilledUp LCD. <http://www.skilledup.com/blog/the-best-mooc-provider-a-review-of-coursera-udacity-and-edx/> Luettu 14.2.2013.

24 Mikä on mooc? Verkkodokumentti. Helsingin Yliopisto Tietojenkäsittelylaitos.
<http://mooc.cs.helsinki.fi/content/mik%C3%A4-mooc>.Luettu 1.2.2013.

25 Welcome to Coursebuilder. 2013. Google Project Hosting. Verkkodokumentti. <https://code.google.com/p/course-builder/> Luettu 14.1.2013.

26 About Kune. Verkkodokumentti. Kune. <http://kune.ourproject.org/about/>. Luettu 5.2.2013.

27 The Most Powerful tools for your classroom. Verkkodokumentti. Blackboard. <https://www.coursesites.com/webapps/Bb-sites-course-creation-BBLEARN/pages/learn.html> Luettu 14.1.2013.

28 Conceptboard for schools and universities. 2013. Verkkodokumentti. Conceptboard. <http://conceptboard.com/for-education>. Luettu 14.1.2013.

29 About Twiddla. 2013. Verkkodokumentti. Expat Software. <http://www.twiddla.com/About/About.aspx> Luettu 14.1.2013

30 Scribblar- Getting Started. 2010. Verkkodokumentti. Muchosmedia Ltd. <http://www.scribblar.com/videos/ >Luettu 14.1.2013.

31 How Stixy works. Verkkodokumentti. Stixy. <http://www.stixy.com/welcome>. Luettu 14.1.2013.

32 Write code, test code and learn to program. 2013. Verkkodokumentti. Complir Inc. <https://compilr.com/>. Luettu 16.1.2013.

33 About. 2013. Verkkodokumentti. LearnPython.Org <http://www.learnpython.org/about> Luettu 5.2.2013.

34 Courses / Learn Python the hard way. 2012. Verkkodokumentti. Shavian Publishing, LLC. <https://inculcate.me/school/courses/2/> Luettu 2.2.2013.

35 About Codingbat. Verkkodokumentti. CodingBat <http://codingbat.com/about.html>. Luettu 2.2.2013.

36 A bit more about Learnstreet. 2013.Verkkodokumentti. Learnstreet Inc. <http://www.learnstreet.com/about/contact>. Luettu 5.2.2013.

37 Welcome to Learnerstv.com. 2012. Verkkodokumentti. LearnersTv. <http://www.learnerstv.com/> Luettu 5.2.2013.

38 Frequently Asked Questions of ShowMeDo. 2010. Verkkodokumentti. Showmedo. <http://showmedo.com/faq>. Luettu 14.2.2013.

39 About OCW. 2012. Verkkodokumentti. Massachusetts Institute of Technology. <http://ocw.mit.edu/index.htm> Luettu 17.2.2013.

40 Python. 2013. Verkkodokumentti. Envato.
<http://net.tutsplus.com/category/tutorials/python-tutorials/> Luettu 5.2.2013.

41 Laiho, Antti. 2013. Päätoiminen tuntiopettaja, mediatekniikan koulutusohjelma. Metropolia Ammattikorkeakoulu. Keskustelu 28.1.2013.

42 About thenewboston. Verkkodokumentti. Roberts, Bucky.
<http://thenewboston.org/about.php>. Luettu 5.1.2013.

43 What We Do.Verkkodokumentti. Codeacademy.
<http://www.codecademy.com/about>. Luettu 5.1.2013.

44 A Small Intro. Verkkodokumentti. Canvasdropr tm.
<http://www.canvasdropr.com/app> Luettu 18.1.2013.

45 Canvasdropr. 31.7.2012. Verkkodokumentti. CrunchBase.
<http://www.crunchbase.com/company/canvasdropr-2> Luettu18.1.2013.

Feedback needed! 2.4.2013 Minna Maltari

Name: _____ **Class:**

_____

Scale:	1 = Poor	2 = Fair	3 = Good	4 = Very Good	5= Excellent
	0 = Less important				
	5 = Very important				

1. Did you find the video tutorials combined with self-practicing exercises in Codeacademy useful?

Rate on a scale from 1 to 5 and give some feedback: __

*Rate also the importance of this feature in sense of learning: __

2. Could the theory lectures be replaced with videos and self-practicing in teaching of programming?

Rate on a scale from 1 to 5 and give some feedback: __

*Rate also the importance of this feature in sense of learning: __

3. Was the Todo site easy and useful?

Rate on a scale from 1 to 5 and give some feedback: __

*Rate also the importance of this feature in sense of learning: __

4. How would you rate the video tutorials? Are they easy to follow?

Rate on a scale from 1 to 5 and give some feedback: __

*Rate also the importance of this feature in sense of learning: __

5. How was the coding exercising website Codecademy?

Rate on a scale from 1 to 5 and give some feedback: __

*Rate also the importance of this feature in sense of learning: __

6. Did you find the collaborative tool Canvasdrop useful?

Rate on a scale from 1 to 5 and give some feedback: __

*Rate also the importance of this feature in sense of learning: __

7. Should there be home assignments to test what you have learned?

*Rate the importance of this kind of feature in sense of learning: __

Openfeedback➔