



Tomi Laine

Python Essentials opintojakson toteuttaminen MooC-opintojaksona Metropolian Moodleen

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tieto- ja viestintäteknikka

Insinööriyö

1.6.2024

Tiivistelmä

Tekijä:	Tomi Laine
Otsikko:	Python Essentials opintojakson toteuttaminen MooC-opintojaksona Metropolian Moodleen
Sivumäärä:	20 sivua
Aika:	1.6.2024
Tutkinto:	Insinööri (AMK)
Tutkinto-ohjelma:	Tieto- ja viestintätekniikka
Ammatillinen pääaine:	Ohjelmistotuotanto
Ohjaajat:	Janne Salonen

Tämän lopputyön tavoitteena oli dokumentoida Python Essentials -kurssin luominen Metropolian Moodleen. Luodun kurssin tarkoituksena oli tarjota opiskelijoille vaihtoehto Python-ohjelmointikielen opintoihin verkkomuodossa.

Kurssi keskittyy aloittelijoihin, joilla ei ole aiempaa ohjelmointikokemusta. Se on suunniteltu tarjoamaan jäsenelty ja looginen oppimispolku Pythonin parissa. Kurssin sisältö käsittää kahdeksan moduulia, jotka suoritettuaan opiskelija kykenee soveltamaan taitojaan työelämässä, sekä jatkamaan itsenäistä opiskelua myös kurssin loputtua.

Avainsanat: NetAcad, Moodle, Python Essentials

Tämän opinnäytetyön alkuperä on tarkastettu Turnitin Originality Check -ohjelmalla.

Sisällys

Lyhenteet

1	Johdanto	1
2	Miksi juuri Python?	2
2.1	Pythonin historia ja kehitys	2
2.2	Pythonin ominaisuudet ja edut	5
3	Cisco Networking Academy & Moodle LMS	9
3.1	Cisco Networking Academy	9
3.2	Moodle LMS	11
4.	Python Essentials opintojakson toteuttaminen Moodleen	13
4.1	Suunnitteluvaihe	13
4.2	Kurssin rakenne ja sen sisällön kehittäminen	14
4.3	Kurssin testaus ja työmäärän arviointi	16
5	Yhteenveto	17
5.1	Lopputulokset	17
5.2	Kehityskohteet	18
	Lähteet	20

Lyhenteet

- Algoritmi:** Joukko sääntöjä tai ohjeita, joita noudatetaan tietyn ongelman ratkaisemiseksi. Algoritmit ovat keskeisiä ohjelmoinnissa ja tietojenkäsittelytieteessä.
- CNE:** Cisco Networking Academy. Cisco Systemsin perustama maailmanlaajuinen koulutusohjelma, joka tarjoaa esimerkiksi IT- ja verkostoitumiskursseja sekä sertifikaatteja.
- IT:** Informaatio ja tietotekniikka. Laaja ala, joka käsittää tietokoneiden, tietojärjestelmien ja ohjelmistojen käytön.
- LMS:** Oppimisympäristön hallintajärjestelmä (Learning Management System). Alusta, jota käytetään koulutussisältöjen hallintaan, jakamiseen ja seurannan toteuttamiseen.
- Moduuli:** Itsenäinen ohjelmakomponentti. Pythonissa moduuli on tiedosto, joka sisältää Python-koodia, kuten funktioita, luokkia ja muuttujia.
- MOOC:** Massiivinen avoin verkkokurssi (Massive Open Online Course). Verkkopohjainen kurssi, joka on avoin suurelle yleisölle ja mahdollistaa itsenäisen oppimisen.
- Ohjelmointiparadigma:** Tapa ajatella, tai mallintaa ohjelmointitehtävän ratkaisu. Eli tyyli luoda tietokoneohjelma.
- QUIZ:** Testi, jolla mitataan opiskelijan osaamista tietystä aiheesta tai moduulista.
- Syntaksi:** Viittaa ohjelmointikielen sääntöihin ja rakenteisiin, jotka määrittävät kuinka ohjelmakoodi tulisi kirjoittaa. Eli ohjelman ”kielioppi”.

1 Johdanto

Tämän lopputyön tarkoituksena oli dokumentoida Cisco Networking Academyn kurssit **Python Essentials 1** sekä **Python Essentials 2** suunnittelu ja toteutus Moodlen oppimisalustalle.

Python on yksi suosituimmista ohjelmointikielistä ja sen osaaminen on tärkeää esimerkiksi automatisoinnissa, data-analytiikassa, web-kehityksessä tai tekoälyn kehittämisessä. Tämän vuoksi Metropolia halusi tarjota opiskelijoilleen mahdollisuuden oppia Pythonin perusteet helposti ja joustavasti myös Moodleen toteutettuna.

Päädyttiin siis Metropolian toimesta yhdistää Cisco Network Academyn kurssit Python Essentials 1 ja Python Essentials 2 yhdeksi omaksi kurssikseen.

Tämän dokumentin tuottajan vastuulle jäi luoda Metropolialle oma kurssi, Python Essentials, joka pohjautuu Cisco Network Academyn kursseihin.

Cisco Networking Academy, jota tästä lähtien kutsumme myös lyhenteellä CNE, on maailmanlaajuinen koulutusohjelma, jonka Cisco Systems perusti vuonna 1997 [1]. Sen tavoitteena on tarjota opiskelijoille esimerkiksi IT- ja verkostoitumiskursseja. Näihin kursseihin kuuluvat myös lopputyöhön pohjautuvat **Python Essentials 1** ja **Python Essentials 2**.

CNE on saavuttanut merkittävää menestystä. Sen kautta onkin koulutettu miljoonia opiskelijoita ympäri maailmaa. Menestykseen saattaakin liittyä kursseihin liittyvät sertifikaatit eli todistukset. [2.]

Lopputyön tavoitteena oli suunnitella ja toteuttaa Python Essentials -kurssi Metropolian käyttämään oppimisympäristö Moodleen. Kyseinen kurssi pohjautuu CNE:n omiin vastaaviin kursseihin Python Essentials 1 ja 2.

Uusi Moodleen luotava kurssi tulee koostumaan kahdeksasta moduulista, eli aihealueesta. Nämä moduulit yhdessä luovat pohjan Pythonin perusteiden oppimiselle. Jokaisen moduulin kohdalla käydään läpi aihe, tavoite sekä tehtävät. Itse laajempi materiaali kuitenkin opiskellaan CNE:n omalta kurssia vastaavalta moduulilta. Materiaalit läpikäytyään opiskelija tekee Metropolian Moodlella vielä moduulien testiosuuden, jonka perusteella kurssin arviointi syntyy.

Tämän lopputyön toteutettu ja dokumentoitu kurssi on suunnattu etenkin opiskelijoille, joilla ei ole juuri aiempaa kokemusta Pythonilla ohjelmoinnista. Tavoitteena olikin luoda selkeä ja kattava opintojakso, joka kattaa Pythonin perusteet sekä auttaa opiskelijoita kehittämään käytännön ohjelmointitaitoja. Kurssi sisältää teoreettista materiaalia ja testejä ”quizzes”, joiden avulla opiskelijat voivat testata oppimaansa tietoa.

2. Miksi juuri Python?

Python on yksi maailman suosituimmista ja eniten käytetyistä ohjelmointikielistä. Sen suosion taustalla on useita syitä, kuten helppokäyttöisyys, tehokkuus ja monipuolisuus. Tässä osiossa tarkastellaan Pythonin historiaa, ominaisuuksia, käyttökohteita sekä syitä siihen, mikä tekee siitä niin erinomaisen ohjelmointikielen.

2.1 Pythonin historia ja kehitys

Pythonin ohjelmointikielen kehittäminen alkoi 1980-luvun lopulla hollantilaisen ohjelmoijan Guido van Rossumin toimesta. Hän työskenteli aiemmin ABC-ohjelmointikielen parissa, mikä vaikutti suuresti Pythonin syntyyn. [3.] ABC oli yksinkertainen ja helposti opittava kieli, mutta se ei saavuttanut laajaa suosiota. ABC:n kehityksen aikana Van Rossum huomasi kuitenkin mahdollisuuden luoda uusi ohjelmointikieli, joka yhdistäisi helppokäyttöisyyden ja monipuolisuuden. Näin alkoi Pythonin tarina. Vuonna 1991 Python julkaistiin ensimmäisen kerran, sekä sen kehitys on ollut siitä lähtien jatkuvaa. [3.]

Pythonin varhaiset versiot keskittyivät erityisesti yksinkertaiseen syntaksiin ja selkeyteen, mikä tekikin siitä suosittua erityisesti opetusikäisissä ja aloittelijoiden keskuudessa. Python oli suunniteltu tukemaan ohjelmoijien tuottavuutta tekemällä koodista helppolukuista, sekä nopeasti opittavaa.

Pythonin varhaisvaiheet ja 1990-luku

Pythonin varhaiset versiot, kuten 1.0 olivat yksinkertaisia mutta tehokkaita. Ne sisälsivät perustoimintoja, kuten dynaamisen tyyppityksen, automaattisen muistinhallinnan ja korkean tason tietorakenteet – kuten listat. Python erottuikin edukseen, jonka vuoksi sitä alettiin käyttämään laajasti esimerkiksi Unix-pohjaisissa järjestelmissä.

Pythonin ensimmäisissä versioissa oli jo silloin perustavanlaatuisia ominaisuuksia, jotka tekevät siitä suosittua vielä nykyäänkin. Esimerkkinä on Pythonin interaktiivinen tulkki, joka mahdollisti koodin suoran suorittamisen ja testaamisen. Tämä teki ohjelmoinnista intuitiivista ja kokeilukeskeistä, mikä helpotti sen oppimista ja nopeutti ohjelmistojen kehitystä.

Python 2.0 ja kasvu

Vuonna 2000 julkaistiin Python 2.0. Tähän versioon lisättiin monia uusia ominaisuuksia, kuten roskienkeruu (garbage collection), mikä paransi ohjelmien muistinhallintaa ja vakautta. Python 2.0 toi myös käyttöön esimerkiksi list comprehension -syntaksin, joka mahdollisti listojen ja muiden tietorakenteiden luomisen lyhyemmällä ja selkeämmällä koodilla. [3.] Tämä versio vakiinnutti Pythonin asemaa monipuolisena kielenä, jota käytetäänkin laajasti erilaisissa ohjelmoinnin tehtävissä, kuten web-kehityksessä, tietojenkäsittelyssä ja automaatiassa.

Python 2.0 kielen myötä alkoi sen suosio kasvamaan nopeasti, mikä houkutteli kehittäjiä erityisesti sen modularisuuteen ja laajan kirjastoekosysteemin vuoksi. Vuonna 2003 Pythonista tuli yksi viidestä virallisesta kielestä Googlen

projekteissa, mikä lisäsi sen näkyvyyttä ja käyttöä myös suurissa ohjelmisto- ja palveluhankkeissa.

Python 3.0 ja merkittävä uudistus

Vuonna 2008 julkaistiin Python 3.0, joka oli suuri muutos Pythonin historiassa. Python 3.0x -sarjan versiot eivät olleet täysin taaksepäin yhteensopivia Python 2.x -sarjan kanssa [3], mikä aiheutti hämmennystä ja kritiikkiä. Python 3:n tavoitteena oli kuitenkin korjata Python 2:n rakenteellisia ongelmia ja parantaa kielen johdonmukaisuutta. Tämä sisälsi merkittäviä parannuksia muun muassa Unicode-tukeen, mikä oli tärkeää kielen kansainväliselle käytölle ja erityisesti web-sovelluksille.

Yksi suurimmista muutoksista oli Python 3:n siirtyminen selkeämpään ja johdonmukaisempaan tulostusfunktioon (print), joka muuttui funktioksi aiemman avainsanan sijaan [4]. Lisäksi Python 3 toi parannuksia tietorakenteiden käsittelyyn sekä paransi kielen luettavuutta entisestään. Tämä pienensi virheiden mahdollisuutta. Vaikka Python 3 toi mukanaan paljon uusia mahdollisuuksia, jatkoivat monet organisaatiot Python 2:n käyttöä vielä pitkään siirtymävaiheen vuoksi.

Pythonin nykytilanne ja jatkuva kehitys

Nykyään Python on yksi maailman suosituimmista ohjelmointikielistä. Sen käyttö kasvaa jatkuvasti eri aloilla, kuten data-analytiikassa, tekoälyn kehityksessä, web-kehityksessä ja automaatiossa. Pythonin suosio perustuukin sen laajaan ja aktiiviseen yhteisöön, joka kehittää jatkuvasti uusia kirjastoja ja työkaluja, jotka helpottavat ohjelmointiin liittyviä tehtäviä. Esimerkiksi TensorFlow ja PyTorch ovat tekoälyn ja koneoppimisen kirjastoja, jotka ovat suosittuja Pythonin käyttäjien keskuudessa [5].

Pythonin kehitystä ohjaa yhteisö ja Python Software Foundation (PSF), joka hallinnoi kielen tulevaisuuden suuntaviivoja ja julkaisuaikatauluja. Joka vuosi Python kehittyy uusilla ominaisuuksilla, joilla pyritään parantamaan sen

suorituskykyä, turvallisuutta sekä kehittäjien kokemusta. Esimerkiksi Pythonin viimeisimmissä versioissa on keskitytty lisäämään monisäikeisyyden tukea, optimoimaan suorituskykyä ja kehittämään uusia työkaluja, kuten kehittäjille parempia virheenkäsittelymekanismeja.

Pythonin tulevaisuus yhtenä maailman tärkeimmistä ohjelmointikielistä näyttää valoisalta. Sen joustavuus ja helppokäyttöisyys tekevät siitä houkuttelevan vaihtoehdon niin aloittelijoille kuin kokeneillekin ohjelmoijille. Yhteisön jatkuva tuki ja uusien ominaisuuksien kehitys varmistavat, että Python pysyy relevanttina ja kilpailukykyisenä myös tulevaisuudessa.

2.2 Pythonin ominaisuudet ja edut

Pythonin suosio perustuu useisiin keskeisiin ominaisuuksiin, jotka tekevät siitä erinomaisen valinnan. Seuraavaksi käymme näistä muutamia läpi:

Helppokäyttöisyys ja selkeä syntaksi

Ominaisuus: Pythonin syntaksi on suunniteltu olemaan helposti luettava ja yksinkertainen verrattuna moniin muihin ohjelmointikieliin, kuten C++:aan tai Javaan [6]. Pythonin tarkoituksena onkin helpottaa sekä koodin kirjoittamista että lukemista. Tämä vähentää koodiin tehtyjen virheiden määrää ja tekee ohjelmoinnista yksinkertaisesti mukavampaa.

Etu: Python sopii erityisesti aloitteleville ohjelmoijille. Kielen opiskelu on yksinkertaista, koska kehittäjät voivat keskittyä ratkaisuihin ilman, että heidän tarvitsee kamppailla monimutkaisten sääntöjen kanssa. Kokeneet kehittäjät arvostavat myös koodin helppoa ylläpidettävyyttä ja sen yksinkertaisuutta yhteistyöprojekteissa.

```
1 def tervehdys(nimi):  
2     return f"Hei, {nimi}!"  
3  
4 print(tervehdys("Maija"))
```

Kuva 1. Esimerkki Pythonin syntaksista.

Monipuolisuus ja ohjelmointiparadigmat

Ominaisuus: Python tukee useita ohjelmointiparadigmoja, kuten olio-ohjelmointia, funktionaalista ohjelmointia ja prosessiperusteista ohjelmointia. Tämä tekee siitä erittäin joustavan ohjelmointikielen, joka sopii monenlaisiin projekteihin.

Etu: Monipuolisuus tarkoittaa, että Pythonilla voi toteuttaa lähes minkä tahansa ohjelmistoprojektin aina pienestä skriptistä suuriin järjestelmiin asti. Tämä tekee siitä houkuttelevan vaihtoehdon monialaisissa ympäristöissä, joissa kehittäjät voivat käyttää samaa kieltä erilaisissa tehtävissä. Ohjelmointiparadigmojen hyödyntäminen tekeekin Pythonista ihanteellisen työkalun erilaisten projektien toteuttamiseen.

Laaja kirjastoekosysteemi

Ominaisuus: Pythonilla on yksi maailman laajimmista ja monipuolisimmista kirjastoekosysteemeistä. Sen standardikirjasto kattaa laajan valikoiman toimintoja, mutta tämän lisäksi Pythonilla on laajat kolmannen osapuolen kirjastot. Nämä tekevät siitä tehokkaan työkalun monimutkaisten sovellusten kehittämiseen, olipa kyseessä sitten datankäsittely, web-kehitys, koneoppiminen tai visualisointi. Pythonilla on todennäköisesti aina valmis kirjasto, joka auttaa toteuttamaan projektin vaatimukset.

Etu: Pythonin kirjastojen ansiosta kehittäjien ei tarvitse keksiä pyörää uudelleen. He voivatkin hyödyntää olemassa olevia, jo testattuja ja ylläpidettyjä kirjastoja nopeuttaakseen omaa kehitysprosessia. Tämä vähentää projekteihin liittyvää kehitysaikaa ja kustannuksia merkittävästi.

Alustariippumattomuus

Ominaisuus: Python on alustariippumaton ohjelmointikieli, mikä tarkoittaa, että sen koodi toimii useilla eri käyttöjärjestelmillä ilman suurempia muutoksia. [6.] Näitä käyttöjärjestelmiä ovat esimerkiksi Windows, MacOS ja Linux. Kehittäjät voivat kirjoittaa koodinsa yhdellä alustalla ja suorittaa sen toisella, mikä tekee koodista helposti siirrettävän ja käytettävän eri ympäristöissä.

Etu: Tämä ominaisuus tekee Pythonista erittäin joustavan. Kehittäjät voivat työskennellä eri ympäristöissä ilman, että heidän tarvitsee huolehtia suurista muutoksista koodissa. Monialustaisissa projekteissa Python tarjoaa sujuvan ja nopean siirtymän yhdeltä käyttöjärjestelmältä toiselle, mikä lisää tehokkuutta ja vähentää ohjelmoinnin pullonkauloja.

Vahva yhteisö ja jatkuva tuki

Ominaisuus: Pythonilla on yksi suurimmista ja aktiivisimmista kehittäjäyhteisöistä maailmassa. Tämä tarkoittaa, että ohjelmistokehittäjillä on pääsy valtavaan määrään dokumentaatioita, foorumeita, tukiryhmiä ja verkkokursseja. Pythonin yhteisö on aktiivinen ja he kehittävät jatkuvasti uusia työkaluja, kirjastoja sekä ratkaisuja, jotka parantavat kielen toimivuutta ja tarjoavat kehittäjille uusia työkaluja.

Etu: Yhteisön vahvuus tarkoittaa, että apua ja resursseja on helposti saatavilla. Jos kehittäjä kohtaa ongelman tai kysymyksen, hän voi löytää vastauksia nopeasti ja helposti yhteisöltä. Tämä vähentää ongelmanratkaisuun kuluvaan aikaa ja parantaa kielen käytettävyyttä hankalissakin projekteissa. Yhteisö tarjoaa myös paljon ilmaista oppimateriaalia, jotka tulevat kehittäjiä.

Kattava tuki tekoälylle ja koneoppimiselle

Ominaisuus: Python on vakiintunut kieli tekoälyn ja koneoppimisen sovelluksissa. Se tukee suosittuja koneoppimiskirjastoja, kuten TensorFlow, PyTorch, Scikit-learn ja Keras [5]. Tämä tekee siitä hyvän työkalun sekä tutkimukselliseen että tuotantoon valmiiseen tekoälykehitykseen.

Etu: Pythonin laaja tuki tekoälyssä ja koneoppimisessa antaa kehittäjille mahdollisuuden hyödyntää näitä kirjastoja helposti. Tämä helpottaa myös tutkimusprosessia, sillä Pythonin avulla kehittäjät voivat nopeasti toteuttaa ja testata uusia malleja ja algoritmeja.

Pythonin interaktiivisuus ja prototyyppien nopea kehittäminen

Ominaisuus: Python tukee interaktiivista komentotulkkiä, jonka avulla ohjelmoijat voivat kirjoittaa ja testata koodia välittömästi. Tämä tekee Pythonista erityisen sopivan nopeaan koodin kehitykseen, sillä koodia voi ajaa ja muokata välittömästi ilman erillistä kääntämisvaihetta.

Etu: Interaktiivinen komentotulkki nopeuttaa sovellusten kehittämistä ja virheiden korjausta, koska ohjelmoijat voivat testata ja tarkistaa koodia reaaliajassa.

Laaja tuki tietokäsittelylle ja datatieteelle

Ominaisuus: Python on johtava ohjelmointikieli datankäsittelyssä ja datatieteessä. Sen laajat kirjastot, kuten Pandas, NumPy ja Matplotlib tarjoavat tehokkaita työkaluja suurten datamäärien analysointiin, visualisointiin ja käsittelyyn.

Etu: Pythonin datan käsittelyyn liittyvät ominaisuudet tekevät siitä suosittun työkalun tutkijoille ja datatieteilijöille. Pythonin avulla he voivat analysoida suuria tietomääriä nopeasti ja tehokkaasti, mikä nopeuttaa dataan perustuvien päätösten tekemistä ja uusien mallien kehittämistä.

3. Cisco Networking Academy ja Moodle LMS

Cisco Networking Academy ja Moodle LMS ovat molemmat merkittäviä tekijöitä nykyaikaisessa verkko-opetuksessa, mutta erityisesti IT-alan taitojen opetuksessa. Yhdessä nämä kaksi järjestelmää tarjoavat opiskelijoille ja kouluttajille laajan valikoiman työkaluja, joiden avulla varmistetaan, että opiskelijat oppivat tehokkaasti ja saavuttavat käytännön taitoja. Näitä opittuja taitoja he pystyvät myöhemmin soveltamaan työelämässä. Tämä osio tarkastelee yksityiskohtaisesti sekä Cisco Networking Academy ohjelman että Moodle LMS -järjestelmän keskeisiä etuja ja piirteitä.

3.1 Cisco Networking Academy

Cisco Networking Academy on yli 20 vuoden ajan ollut yksi maailman johtavista IT-koulutusohjelmista. Ohjelma on suunniteltu tarjoamaan opiskelijoille käytännönläheistä, laadukasta ja kattavaa koulutusta etenkin IT-taidoissa, jotka ovat kriittisiä digitalisoituvassa maailmassa. [1.]

Cisco Networking Academy -ohjelman tavoitteena on antaa kaikille mahdollisuus oppia sekä tulla osaksi globaalisti kasvavaa ammattilaisten joukkoa. Ohjelma on erityisen tärkeä kehittyvissä maissa ja alueilla, joilla mahdollisuudet perinteiseen koulutukseen voivat olla rajalliset.

The cybersecurity job market is hot!
See why 5 million learners began their journey with our [free Intro to Cybersecurity course](#)

Learn the skills, land your dream job.

Free online courses. In-person learning. Certification-aligned pathways in topics like Cybersecurity, Networking, and Python.

It's all here. Are you ready to begin, change, or propel your career?

[Start Learning](#) [Explore Subjects](#)

Kuva 2. Cisco Networking Academyn oppimisympäristö.

Keskeiset piirteet ja edut:

Modulaarisuus ja joustavuus: Cisco Networking Academyn kurssit on usein jaettu moduuleihin, jotka keskittyvät tiettyihin osaamisalueisiin. Juuri tämä modulaarisuus mahdollistaa joustavan oppimisen, jossa opiskelijat voivat edetä omaan tahtiinsa ja valita itselleen tärkeimmät opintojen osa-alueet.

Käytännönläheinen oppiminen: Ohjelman ytimessä on käytännön taitojen kehittäminen. Cisco Networking Academy tarjoaa opiskelijoille mahdollisuuden tehdä harjoituksia, simulaatioita ja virtuaalilaboratorioita, joissa opitaan esimerkiksi verkkokonfigurointia tai kyberturvallisuuden perusteita. Tämä auttaa oppilaita soveltamaan teoriaa konkreettisissa tilanteissa, mikä valmistaa heidät työelämän tuleviin haasteisiin.

Interaktiivisuus ja simuloinnit: Cisco Networking Academy käyttää laajasti interaktiivisia työkaluja opintojen tueksi, kuten esimerkiksi Cisco Packet Tracer -simulaattoria, joka mahdollistaa verkkoinfrastruktuurien suunnittelun ja testauksen ilman fyysisiä laitteita. Tämä tarjoaa opiskelijoille mahdollisuuden harjoitella ja ymmärtää, miten todelliset verkot toimivat ilman omia kalliita laitteistohankintoja.

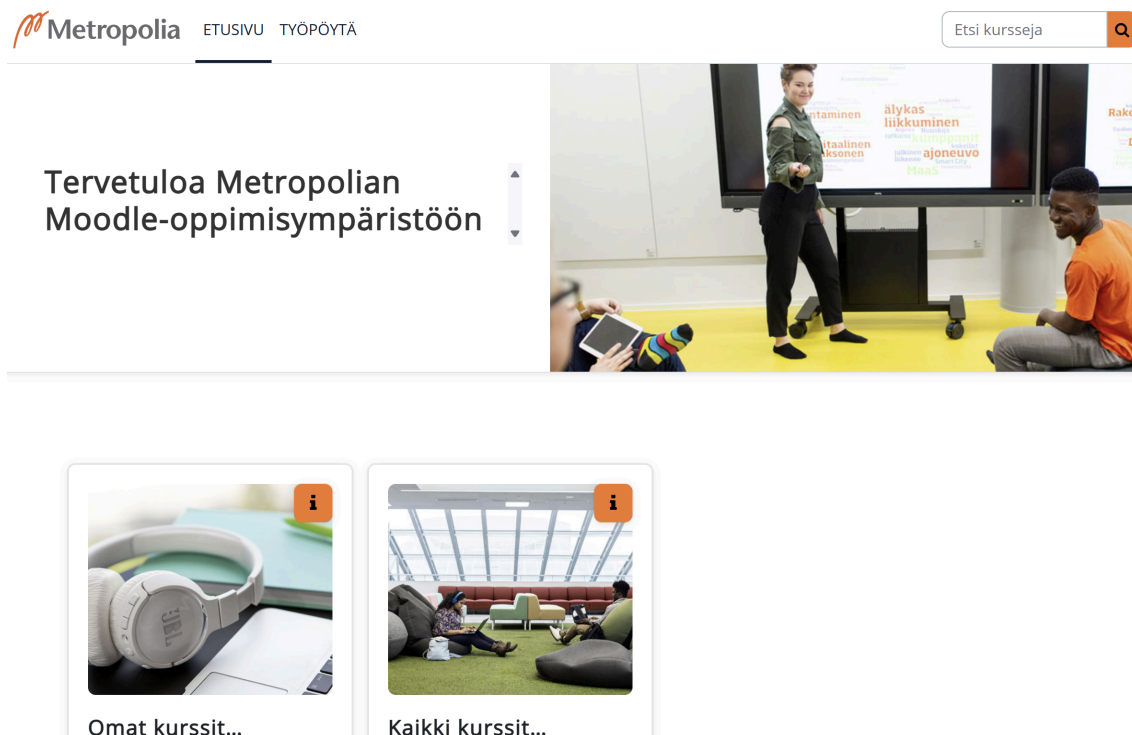
Sertifiointit: Yksi ohjelman parhaita puolia on mahdollisuus suorittaa kansainvälisesti tunnettuja Cisco-sertifikaatteja, kuten CCNA (Cisco Certified Network Associate). Sertifikaatit ovat arvostettuja, sekä parantavat työnhakijoiden asemaa työmarkkinoilla. Ne osoittavat, että henkilö hallitsee tarvittavat tekniset taidot ja osaa soveltaa niitä myös käytännössä.

Yhteisöllisyys ja verkostoituminen: Cisco Networking Academy rohkaisee opiskelijoita osallistumaan mukaan globaaliin IT-yhteisöön. Ohjelman avulla opiskelijat voivat verkostoitua muiden oppilaiden ja IT-ammattilaisten kanssa, mikä voi avata ovia tuleviin työllistymismahdollisuuksiin. Yhteisöllisyys tukee myös oppimista, sillä opiskelijat voivat jakaa kokemuksiaan ja auttaa toisiaan vaikeiden konseptien omaksumisessa.

Tulevaisuuden suuntaukset ja kehityskohteet: Cisco Networking Academy pyrkii jatkuvasti vastaamaan työmarkkinoiden muuttuviin tarpeisiin. Tulevaisuudessa ohjelman odotetaan laajenevan vastaamaan tekoälyn, pilvipalveluiden ja tietoturvan haasteisiin. Näin opiskelijat ovat aina ajan tasalla uusimmista teknologioista ja markkinoiden vaatimuksista.

3.2 Moodle LMS

Moodle LMS (Learning Management System) on maailmanlaajuisesti käytetty avoimen lähdekoodin oppimisympäristö [7], jota hyödynnetään niin perus- kuin jatkokoulutuksessakin. Moodlen avoin lähdekoodi mahdollistaa sen joustavan räätälöinnin erilaisiin opetustarpeisiin. Se on erinomainen valinta sekä yksilölliseen että laajempaan koulutuskokonaisuuteen.



Kuva 3. Moodle LMS -oppimisympäristö.

Moodlen keskeiset piirteet ja edut:

Käyttäjätavallinen käyttöliittymä: Moodlen käyttöliittymä on suunniteltu helppokäyttöiseksi sekä opiskelijoille että opettajille. Käyttäjät voivat helposti navigoida eri kurssien, materiaalien, tehtävien sekä tenttien välillä. Opettajille on tarjolla hallintatyökaluja kurssien luomiseen, opiskelijoiden arviointiin ja oppimisen seurantaan. [8.]

Monipuoliset oppimateriaalit ja aktiviteetit: Moodle tukee monenlaisia oppimateriaaleja, kuten videoita, äänitiedostoja, dokumentteja sekä muita interaktiivisia aktiviteetteja. Opettajilla on myös mahdollisuus hyödyntää esimerkiksi Moodlen keskustelufoorumeita. [8.] Nämä mahdollistavat monimuotoisen ja rikkaan oppimisympäristön, jossa oppijat voivat hyödyntää erilaisia oppimistylejä.

Integraatio muiden järjestelmien kanssa: Moodleen voidaan asentaa laajennuksia “plugins” sekä näin integroida se useiden eri järjestelmien ja

palveluiden, kuten Zoomin tai Google Workspacen kanssa. [8.] Tämä yhdistää eri digitaaliset välineet opiskelun tueksi.

Joustavat arviointi- ja seurantatyökalut: Moodlella on laajat arvioinnin mahdollistavat ominaisuudet, jotka mahdollistavat opiskelijoiden edistymisen seurannan. Opettajat voivat luoda esimerkiksi testejä tai kyselyitä, jotka tarjoavat yksityiskohtaista palautetta opiskelijoille. [8.]

Turvallisuus ja yksityisyydensuoja: Moodle on suunniteltu vastaamaan tiukkoihin turvallisuusstandardeihin, erityisesti opiskelijoiden tietojen osalta. Koulutusorganisaatioille, jotka käsittelevät arkaluontoista tietoa, kuten henkilökohtaisia tietoja, Moodlen turvallisuusominaisuudet ovat keskeisiä.

Moodle tarjoaa monia mahdollisuuksia verkko-oppimisen ja hybridimallin toteutukseen. Pandemian aikana etäopetus kasvoi merkittävästi ja Moodle onkin kasvanut avainasemaan tarjoamalla tehokkaita työkaluja digitaaliseen oppimiseen. Niiden avulla opettajat voivat luoda vuorovaikutteisia kursseja, jotka tekevät oppimisprosessista dynamisempää ja kiinnostavampaa.

4. Python Essentials -opintojakson toteuttaminen Moodleen

4.1 Suunnitteluvaihe

Python Essentials opintojakson suunnittelu Metropolian Moodle-oppimisympäristöön alkoi tavoitteiden asettamisella. Projektin alussa käytiin läpi ohjaaja Janne Salosen kanssa kurssin tavoitteet sekä kaikki, mitä tulevan kurssin tulisi sisältää.

Tavoitteeksi asetettiin Python ohjelmoinnin kurssin luominen, joka perustuu Cisco Networking Academyn kursseihin nimeltään Python Essentials 1 ja 2. Tämän kurssin tulisi tarjota ohjelmoinnin alkeita opiskelijoille, joilla ei ole aiempaa ohjelmointikokemusta. Kurssin tulisi myös varmistaa, että opiskelijat

saavuttavat riittävän osaamistason kyetäkseen soveltamaan oppimaansa työelämässä.

Alkuvaiheessa kartoitettiin Moodle-alustan tarjoamat mahdollisuudet ja haasteet. Tarkoituksena oli hyödyntää Moodlea niin, että kurssi olisi mahdollisimman interaktiivinen ja kiinnostava opiskelijan kannalta. Tämä edellytti alustaan tutustumista sekä sen työkalujen että ominaisuuksien tutkimista, jotta saataisiin luotua hyvä verkkokurssi.

4.2 Kurssin rakenne ja sen sisällön kehittäminen

Python Essentials -opintojakson rakenne on suunniteltu tarjoamaan opiskelijoille jäsenelty ja looginen oppimispolku Pythonin ohjelmoinnin perusteiden ymmärtämiseksi. Kurssi on jaettu kahdeksaan moduuliin, jolla varmistetaan se, että opiskelijat voivat rakentaa osaamistaan askel askeleelta:

Module 1. Introduction to Python and Computer Programming

Module 2: Data Types, Variables and Basic I/O Operations

Module 3. Boolean Values, Conditional Execution, Loops, Lists and List Processing, Logical and Bitwise Operations

Module 4. Functions, Tuples, Dictionaries, Exceptions and Data processing

Module 5. Modules, Packages and PIP

Module 6. Strings, String and List Methods, Exceptions

Module 7. Object-Oriented Programming

Module 8. Miscellaneous

Jokainen näistä moduuleista koostuu kahdesta osasta: teoreettisesta ja käytännön osiosta.

Teoreettisessa osiossa opiskelijat perehtyvät moduulien keskeisiin käsitteisiin ja teoriaan, joka antaa heille tarvittavat tiedot tulevia käytännön harjoituksia varten. Tämä osio sisältää moduulipohjaisesti yksityiskohtaista teoriaa ohjelmoinnin peruskäsitteistä. Jokainen aihe käsitellään perusteellisesti. Lisäksi niihin liittyy myös esimerkkejä, jotka auttavat opiskelijoita ymmärtämään käsitteet käytännön sovelluksissa.

Teoreettinen osio ei ole pelkkää tiedon jakamista, vaan se tarjoaa myös pohjan ongelmanratkaisukyvyyn kehittämiseksi. Tämän vuoksi oppimateriaaliin sisällytetään usein myös esimerkiksi kaavioita, diagrammeja ja havainnollistavia animaatioita, jotka konkretisoivat teoriassa käsitellyjä aiheita.

Käytännön osio puolestaan tarjoaa opiskelijoille mahdollisuuden soveltaa oppimaansa tekemällä harjoitustehtäviä ja kokeita, jotka on suunniteltu vahvistamaan teoriaosuuden oppeja.

Kurssimateriaalit on luotu mahdollisimman helposti ymmärrettäviksi. Tämä tarkoittaa, että materiaalit sisältävät selkeitä ja vaiheittaisia oppeja, jotka tukevat opiskelijoita itsenäisessä opiskelussa. Esimerkiksi vaiheittaiset ohjeet havainnollistavat keskeisiä ohjelmointikäsitteitä, sekä ne on varustettu visuaalisilla elementeillä, kuten kaavioilla tai kuvilla, jotka helpottavat käsitteiden ymmärtämistä. Lisäksi kurssille on luotu interaktiivisia elementtejä, kuten testit, ”quizzes”, jotka tarjoavat opiskelijoille mahdollisuuden testata omaa osaamistaan ja saada siitä välitöntä palautetta.

Jokaisen moduulin teoria ja käytännön osion suoritettuaan käy opiskelija läpi testin ”quiz”, jonka läpäistyään opiskelija voi siirtyä jatkamaan seuraavaan moduuliin.

Nämä testit ovat noin 10-20 kysymyksen mittaisia, jotka mittaavat opiskelijan ymmärrystä moduulin teoriaosuuden ja käytännön osion sisällöstä. Opiskelijan

tulisi saada jokaisesta quizzista vähintään 70 % kysymyksistä oikein. Aikaa yhden testin yrityksen tekoon on tunti ja samaa testiä saa yrittää tehdä kolmesti.

Testejä varten on luotu Moodleen jokaiselle moduulille kysymyspankki, joista kysymykset valitaan satunnaisjärjestyksessä jokaiselle kokeen yrityskerralle. Näin ollen jokainen kokeen yritys on erilainen.

4.3 Kurssin testaus ja työmäärän arviointi

Ennen kurssin julkaisemista käytiin kurssi vielä läpi ja varmistettiin sen toimivuus sekä käytettävyys. Tämä sisälsi kurssin moduulien suorittamisen sekä niiden tarkastelun. Testauksen aikana arvioitiin muun muassa, ovatko tehtävät ja ohjeet selkeitä ja ymmärrettäviä.

Kurssi käytiin läpi kokonaisuudessaan myös opiskelijan näkökulmasta. Tämä sisälsi kaikkien moduulien läpikäymisen ja tehtävien suorittamisen sekä kurssin kokeiden testauksen. Tavoitteena oli varmistaa, että kurssin käyttö olisi suhteellisen ongelmaton.

Tavoitteena oli myös varmistaa, että kurssi ei olisi liian kuormittava opiskelijoille, mutta tarjoaisi silti riittävän haasteen oppimisen edistämiseksi. Tämä sisälsi arvioinnin siitä, kuinka paljon aikaa opiskelijat tarvitsevat tehtävien suorittamiseen sekä kuinka hyvin kurssin sisällöt vastaavat opiskelijoiden oppimisnopeutta.

Lisäksi käytiin läpi kurssin materiaalien monipuolisuutta ja soveltuvuutta eri oppimisvaiheisiin. Tämä tarkoittaa, että kurssi ei ole vain haasteellinen, mutta tarjoaa myös riittävästi tukea ja resursseja opiskelijoiden oppimisprosessin tukemiseksi.

Kurssin tekniset komponentit, kuten linkkien toimivuus ja interaktiiviset elementit tuli myös käydä läpi, jotta kaikki toimisivat suunnitellusti.

5. Yhteenveto

5.1 Lopputulos

Tässä työssä on käsitelty Python Essentials -opintojakson suunnittelua ja toteutusta Metropolian Moodle-oppimisympäristöön. Projekti sai alkunsa Metropolia Ammattikorkeakoulun tarpeesta tarjota opiskelijoille mahdollisuutta oppia Pythonin ohjelmoinnin perusteet helposti ja joustavasti. Python valittiin sen helppokäyttöisyyden, monipuolisuuden ja laajan käytön vuoksi maailmalla.

Kurssin suunnittelu alkoi tavoitteiden asettamisella lopputyön ohjaajan Janne Salosen kanssa. Alkuvaiheessa määriteltiin kohdeyleisöksi opiskelijat, tai henkilöt, joilla ei ole aiempaa kokemusta ohjelmoinnista. Moodlen tarjoamat mahdollisuudet ja haasteet kartoitettiin, jotta kurssi voitaisiin rakentaa toimivaksi.

Kurssin sisältö kehitettiin moduulipohjaiseksi, mikä mahdollisti johdonmukaisen etenemisen perusasioista monimutkaisempiin aiheisiin. Teoreettiset osiot yhdistettiin käytännön harjoituksiin. Lisäksi oppimista tukevat interaktiiviset elementit, kuten kokeet, sisällytettiin kurssiin. Kurssimateriaalit suunniteltiin selkeiksi ja helposti ymmärrettäviksi, jotta ne tukisivat erityisesti aloittelijoita.

Kurssi testattiin ennen julkaisua, sekä sen toimivuus ja käytettävyys varmistettiin. Testauksessa arvioitiin tehtävien ja ohjeiden selkeys, kuten myös kurssin vaadittava työmäärä. Kurssi tarjoaa riittävästi haastetta, mutta ei välttämättä ole erityisen raskas suoritettavaksi.

Kohdeyleisön tarpeet huomioitiin kurssin mukauttamisessa. Kurssi suunniteltiin tukemaan oppimista alkeista alkaen ja se tarjoaa selkeän ja johdonmukaisen polun ohjelmointitaitojen kehittämiseen. Tämä lähestymistapa varmistaa, että opiskelijat pystyvät saavuttamaan kurssin oppimistavoitteet ja jatkamaan opintojaan, tai soveltamaan oppimaansa työelämässä.

Python Essentials 3 op

Kurssi Asetukset Osallistujat Arvioinnit Raportit Lisää ▾

Course Overview About the Course Taking the Course Module 1. Introduction to Python... Module 2. Python Data Types, Va...
 Module 3. Boolean Values, Condi... Module 4. Functions, Tuples, Dict... Module 5. Modules, Packages an... Module 6. Strings, String and List ...
 Module 7. Object-Oriented Progr... Module 8. Miscellaneous



Python Essentials: A Comprehensive Course Overview

Kuva 4. Python Essentials Moodle.

Lopputuloksena oli selkeästi jäsenneilty ja toimiva kokonaisuus, joka kattaa Python ohjelmoinnin perusasiat verkkomuodossa. Kurssi koostuu kahdeksasta moduulista, jotka johdattelevat opiskelijaa loogisesti alkeista edistyneempiin ohjelmointikäsitteisiin. Kurssin käytännönläheisyys ja interaktiiviset elementit, kuten testit ja tehtävät mahdollistavat oppimisen omassa tahdissa.

5.2 Kehityskohteet

Kurssi on saanut hyvän perusmuodon, mutta tulevaisuudessa sen laatua ja toimivuutta voidaan parantaa monin tavoin. Alla on joitakin ehdotuksia kehitysalueista, jotka voisivat parantaa kurssin oppimiskokemusta, käytettävyyttä ja ajankohtaisuutta.

Lisämateriaalin kehittäminen:

Vaikka kurssi sisältää jo perustiedot Pythonilla ohjelmoinnista, voi lisämateriaalin lisääminen rikastuttaa oppimiskokemusta huomattavasti.

Seuraavia lisämateriaaleja voitaisiin käyttää:

Video-oppaat: Video-opetuksen hyödyntäminen on nykyopetuksessa tärkeässä roolissa. Video-oppaat pohjautuen Python-esimerkkeihin voisi helpottaa oppilaiden käsitystä monimutkaisista aiheista. Opiskelijat voisivat seurata esimerkkejä reaaliajassa, jolloin oppiminen olisi dynaamisempaa ja käytännönläheisempää.

Syventävät koodiesimerkit ja tehtävät: Koodiesimerkkien ja monipuolisempien harjoitustehtävien lisääminen syventäisi oppimista. Olisi mahdollista sisällyttää monimutkaisempia ohjelmointiongelmia, jotka voisi liittyä esimerkiksi tietorakenteisiin, algoritmeihin tai eri Python-kirjastojen käyttöön.

Palautemekanismi: Palautemekanismeja voitaisiin kehittää, jotta opiskelijat voivat antaa reaaliaikaista palautetta oppimiskokemuksestaan. Tämä parantaisi kurssin laatua ja auttaisi tunnistamaan alueita, joissa opiskelijat tarvitsevat enemmän tukea.

Kurssin saavutettavuus: Kurssin kääntäminen useille kielille parantaisi sen saavutettavuutta globaalilla tasolla. Voitaisiin myös pohtia, miten materiaali olisi paremmin käytettävissä vaikkapa näkörajoitteisille tai oppimisvaikeuksista kärsiville opiskelijoille. Moodle LMS onkin ansainnut saavutettavuustason (WCAG 2.1 Level AA) tammikuussa 2024 [9]. Tämän tason saavutettavuus tarkoittaa, että Moodle oppimisympäristö täyttää useimmat saavutettavuuden vaatimukset, joka tarkoittaa parempaa saavutettavuutta esimerkiksi näkö tai kuulohaasteisille.

Jatkuva kurssipäivitys ja sisällön kehittäminen: Pythonin kehitys jatkuu koko ajan, minkä vuoksi myös kurssin sisällön tulisi kehittyä. On tärkeää päivittää kurseja säännöllisesti, jotta opiskelijat oppivat ajantasaista ja relevanttia tietoa.

Näiden kehitysehdotusten myötä Python Essentials kurssista voitaisiin luoda entistä monipuolisempi ja saavutettavampi koulutuspaketti, joka vastaa sekä opiskelijoiden että työelämän muuttuviin tarpeisiin myös tulevaisuudessa.

Lähteet

1. NetAcad. About. Verkkoaineisto: [About Cisco Networking Academy \(netacad.com\)](https://www.netacad.com) Luettu 23.09.2024.
2. Cisco. CCNA. Verkkoaineisto: [CCNA - Cisco](https://www.cisco.com) Luettu 23.09.2024.
3. Wikipedia. History of Python. Verkkoaineisto: [History of Python - Wikipedia](https://en.wikipedia.org) Luettu 23.09.2024.
4. Geeksforgeeks. Python-version-history. Verkkoaineisto: [Python Version History - GeeksforGeeks](https://www.geeksforgeeks.org) Luettu 23.09.2024.
5. Unite.Ai. 10 parasta Python-kirjastoa syväoppimiseen. Verkkoaineisto: [10 parasta Python-kirjastoa syväoppimiseen \(2024\) - Unite.AI](https://unite.ai) Luettu 23.09.2024.
6. Geeksforgeeks. Python – The new generation Language. Verkkoaineisto: [Python – The new generation Language - GeeksforGeeks](https://www.geeksforgeeks.org) Luettu 23.09.2024.
7. Moodle Docs. Opettajan Moodle-opas. Verkkoaineisto: [MoodleDocs](https://moodle.org) Luettu 15.09.2024.
8. Moodle Docs. Features. Verkkoaineisto: [Features - MoodleDocs](https://moodle.org) Luettu 15.09.2024.
9. Moodle Docs. Accessibility. Verkkoaineisto: [Accessibility | Moodle Developer Resources](https://moodle.org) Luettu 15.09.2024.
10. Datacamp. What is Python used for? Verkkoaineisto: [What is Python Used For? 7 Real-Life Python Uses | DataCamp](https://www.datacamp.com) Luettu 23.09.2024.
11. Python Software Foundation. About. Verkkoaineisto: [About the Python Software Foundation | Python Software Foundation](https://www.python.org) Luettu 23.09.2024.

