# Ohjelmointitaitojen ja ohjelmoinnillisen ajattelun tukeminen

Opas vuosiluokille 1 – 6

Ville-Juhani Nivasalo

2023

UUDET LUKUTAIDOT

OPETUSHALLITUS RAHOITTAA HANKETTA

Vastuullinen arktinen korkeakouluyhteisö LUC

LAPIN YLIOPISTO
UNIVERSITY OF LAPLAND

LAPIN AMK
Lapland University of Applied Sciences

# Johdanto

Tämä opas on tarkoitettu Lapin yliopiston harjoittelukoulun opettajille, muulle henkilöstölle sekä opiskelijoille ja heidän ohjaajilleen. Opasta voi toki käyttää muuallakin, mutta se on rakennettu nimenomaisesti hyödyntämään harjoittelukoululla käytössä olevaa välineistöä.

Oppaan tavoitteena on tukea järjestelmällistä ohjelmointitaitojen ja ohjelmoinnillisen ajattelun kehitystä koko oppijan peruskoulun ajan. Sen pohjana on käytetty Suomen harjoittelukoulujen Tietostrategian 2022 – 2025 hyvän osaamisen tavoitteita.

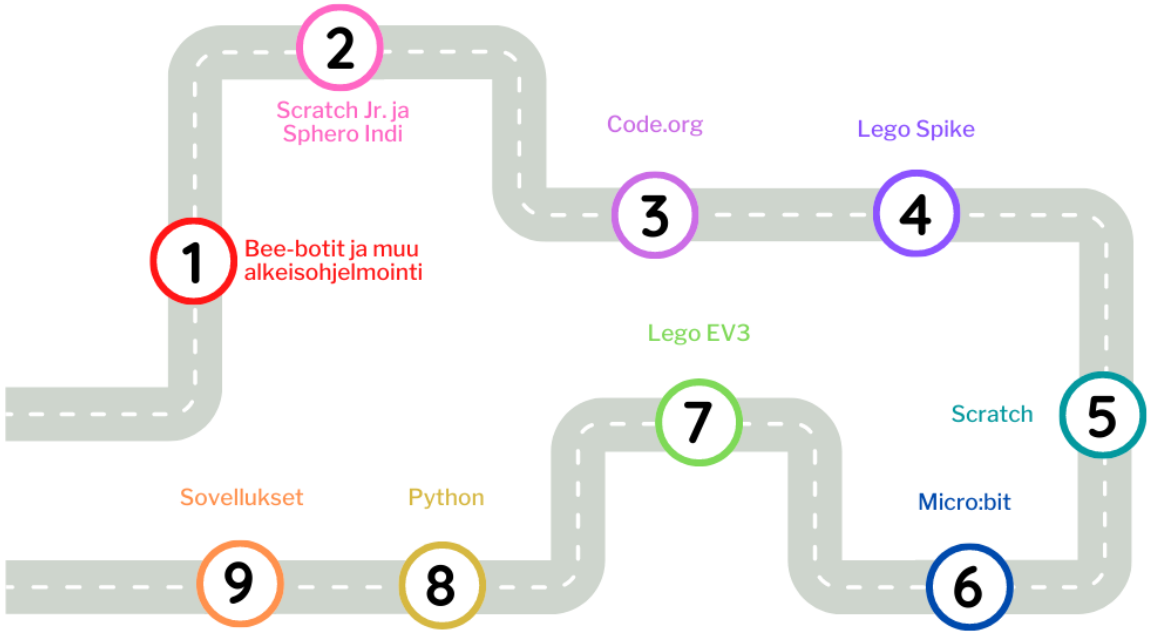
# Tavoitteellinen edistyminen

Oppijan tavoitteiden edistymistä arvioidaan peilaamalla harjoituksia Tietostrategian 2022 – 2025 oppilaan osaamisen kuvauksiin.

Jokaisen vuosiluokan osalta ohjelmointiosaamiseen liittyvät tavoitteet on jaettu tässä oppaassa **yleisiin** ja **erityisiin** tavoitteisiin.

**Yleistavoitteet** ovat kaikkiin oppiaineisiin eheytettäviä ohjelmoinnillisen ajattelun taitoja ja työtapoja. **Erityistavoitteet** vuorostaan ovat sellaisia, joiden harjoitteluun on hyvä käyttää erillisiä ohjelmointialustoja. Juuri näihin keskitytään tässä oppaassa.

Digipassin eri versioita on mahdollista käyttää oppimisen tukena ja arvioinnissa.




# Sisältö

Ohjelmointipolun keskiössä ovat ohjelmoinnillisen ajattelun taidot.

Niiden harjoitteluun on jokaiselle vuosiluokalle valittu ikätasolle sopivia ohjelmointiympäristöjä ja ohjelmoitavia laitteita.

Lisäharjoituksia ja vinkkejä ohjelmointiin ilman laitteita löytyy mm. eNorssin ohjelmointipolulta.

# Huomioita

Tämän dokumentin tavoitteena ei ole olla kaiken kattava tietokirja ohjelmoinnin opettamiseen, vaan sen sijaan toimia yleisesityksenä siitä, miten perusopetuksen oppilaan ohjelmointipolkua voi hahmottaa.

Tästä syystä esityksessä on mukana myös linkkejä sen ulkopuolelle. Ei esimerkiksi olisi ollut tarkoituksenmukaista monistaa Lego Spike tai Sphero Indi –oppaita kaikkia samaan dokumenttiin.

# Vuosiluokat 1 - 2

Oppilas tutustuu ohjelmointiin

# Yleistavoitteet

Ohjelmoinnillinen ajattelu

Oppilas
- osaa järjestää ja vertailla erilaisia asioita eri olosuhteiden ja mallien perusteella
- osaa kertoa havaintojaan käsitteiden ja konkreettisten työkalujen avulla
- osaa jakaa arjen ilmiöihin liittyvät ongelmat pienempiin osiin
- osaa testata erilaisia ratkaisuja ongelmien ratkaisemiseksi
- pystyy kertomaan, miten hän ajatteli

Tutkiva työskentely ja tuottaminen

Oppilas
- oppii tekemään yhteistyötä ryhmässä ja eri rooleja ryhmätyössä

Ohjelmoitujen ympäristöjen tunteminen ja niissä toimiminen

Oppilas
- tunnistaa ja nimeää omasta kokemusmaailmastaan löytyvää tietotekniikkaa ja tutustuu robotiikkaan

# Erityistavoitteet

Ohjelmoinnillinen ajattelu

Oppilas
- osaa laatia leikillisiä ohjeita, jolla hän ohjaa toista opiskelijaa, ohjelmoitavaa laitetta tai yksittäistä hahmoa sovelluksessa
- osaa seurata reseptityylisiä ohjeita (kävele 2 askelta, käänny)

Tutkiva työskentely ja tuottaminen

Oppilas
- osaa esittää omia ideoitaan, kuunnella muiden oppilaiden ideoita ja testata erilaisia ratkaisuja yhdessä muiden kanssa
- osaa ohjattuna luoda tarinoita tai pelejä sisältävän digitaalisen tuotteen animaation tai yksinkertaisen ohjelmoinnin avulla
- käsittelee eri oppiaineiden sisältöä leikillisesti ja kokeellisesti käyttämällä ohjelmointiin liittyviä liiketoimintamalleja ja työkaluja

Ohjelmoitujen ympäristöjen tunteminen ja niissä toimiminen

Oppilas
- tunnistaa ja nimeää omasta kokemusmaailmastaan löytyvää tietotekniikkaa ja tutustuu robotiikkaan
- pohtii, kuinka digitaalisissa ympäristöissä kerätään ja tallennetaan tietoa omasta toiminnastaan
- osaa kertoa ainakin yhden esimerkin siitä, mihin kerättyjä henkilötietoja voidaan käyttää

# Bee-bot

Bee-bot on monelle lapselle tuttu ensimmäinen ohjelmoitava robotti jo päiväkodista.

Vuosien varrella robotin ympärille on kehitetty valtava määrä eheyttäviä harjoituksia eri oppiaineisiin, jotka samalla edesauttavat ohjelmoinnillisen ajattelun kehittymistä.

Esimerkkejä harjoituksista löytyy mm. Luma-keskus Lapin Bee-bot-robotit opetuksessa-materiaalista

# Sphero indi

Indi on Spheron uusimpia, erityisesti alkuopetukseen suunnattuja robotteja. Sille voidaan antaa erilaisia liikkumiseen liittyviä käskyjä lattialle laitettavilla värilohkoilla.

Värilohko-ohjelmointi mahdollistaa syklisen ongelmanratkaisuprosessin, jossa oppilaat voivat joustavasti kokeilla monenlaisia ratkaisuja lyhyessä ajassa. Toisin kuin Bee-botilla, ohjelma ei pyyhkiydy kokonaan pois uutta versiota tehdessä.

Robotti on helppo ottaa haltuun Indi tutuksi –materiaalin sekä siihen liittyvien tukivideoiden avulla.

# Scratch jr

Scratch Junior on alkeisohjelmoinnin harjoitteluun tarkoitettu lohko-ohjelmointisovellus. Ohjelmat kirjoitetaan ja luetaan vasemmalta oikealle liittäen palapelin palojen näköisiä ohjelmointilohkoja toisiinsa.

Auroran Scratch Jr –ohjelmointikoulu on loistava materiaali laitteella tapahtuvan ohjelmoinnin harjoitteluun ja tarinoiden kertomiseen ohjelmoitujen animaatioiden avulla.

Scratch Jr -sovellus löytyy kaikilta Harjoittelukoulun iPadeilta.

Halutessanne voitte käyttää Taitopajan iPad-salkkujen laitteita sekä sieltä löytyviä Apple Pencil –kyniä. Niiden avulla oppilaat voivat piirtää omia hahmojaan ja taustojaan suoraan laitteelle!

# Esimerkkejä vuoden aikana toteutettavista ohjelmointiprojekteista

## 1. luokalla

- Matematiikan oppikirjan harjoitukset
- Bee-bot-materiaali integroituna matematiikkaan
  - Harjoitukset 2, 4, 5, 12, 13, 14
- Sphero indi tutuksi
  - Tehtävät 1 ja 2
- Auroran Scratch Jr. ohjelmointikoulun tehtävät:
  1. Tausta ja hahmo
  2. Ohjelma yhdelle hahmolle (esim. oman animaation/esitelmän/tarinan tekeminen)
- Ohjelmoidun ympäristön havainnointia omassa arjessa

## 2. luokalla

- Matematiikan oppikirjan harjoitukset
- Bee-bot-materiaali integroituna matematiikkaan (symmetria ja peilaaminen)
  - Harjoitukset 7, 8, 9, 10
- Sphero indi
  - Peilaaminen
- Oma Scratch Jr. -projekti

# Lisäksi

- Etsikää ympäristöstä automatisaatioon perustuvia teknologisia ratkaisu (valot, välituntikello, jääkaappi, liikennevalot) ja miettikää, mitä hyötyä on siitä, että ne toimivat automaattisesti. Mitä haittaa siitä voi olla?
- Jutelkaa verkkovälitteisistä mainoksista ja siitä, millaisia kokemuksia oppilailla on niiden automaattisesta kohdentamisesta käyttäjälle. Miettikää myös, mitä tietoja algoritmi eli mainosten taustalla oleva automatiikka tarvitsee kohdentamiseen ja mistä se tuota tietoa saa.
- Pohtikaa tietoturvaa ja sitä, mitä voi tapahtua, jos ei esimerkiksi suojaa omaa puhelintaan turvallisella salasanalla.

# Vuosiluokat 3 - 4

Oppilas harjoittelee ohjelmointia

# Yleistavoitteet

Ohjelmoinnillinen ajattelu

Oppilas
- osaa järjestää, vertailla ja esittää tietoa käsitteiden ja symbolien avulla

Tutkiva työskentely ja tuottaminen

Oppilas
- ymmärtää, että ilmiötä voidaan tarkastella eri näkökulmista ja osaa pohtia muiden näkökulmia

Ohjelmoitujen ympäristöjen tunteminen ja niissä toimiminen

Oppilas
- osaa kertoa esimerkkejä kohdennetuista digitaalisista sisällöistä sekä tavoista, joilla niitä kohdennetaan
- pohtii omaa toimintaansa ja siitä kerääntyvän tiedon käyttöä digitaalisissa ympäristöissä

# Erityistavoitteet

Ohjelmoinnillinen ajattelu

Oppilas
- käyttää erilaisia menetelmiä ja ratkaisumalleja ongelmien ratkaisemiseen ja yrittää myös luoda omia ratkaisuja
- laatii tarkat ja yksityiskohtaiset ohjeet toisto- ja ehtorakenteiden avulla
- etsii ja korjaa ohjeissa ja/tai koodissa olevia virheitä
- käyttää graafista ohjelmointiympäristöä ja osaa luoda siihen oman ohjelman

Tutkiva työskentely ja tuottaminen

Oppilas
- ymmärtää, että ilmiötä voidaan tarkastella eri näkökulmista ja osaa pohtia muiden näkökulmia yhteisessä ohjelmointiprojektissa
- tunnistaa animaatioiden ja pelien ohjelmoinnin piirteitä ja yrittää luoda pelejä erilaisilla animaatio- tai ohjelmointialustoilla
- osaa käyttää ohjelmointiin liittyviä työmenetelmiä ja työkaluja luovaan ilmaisuun ja omaan tuotantoon

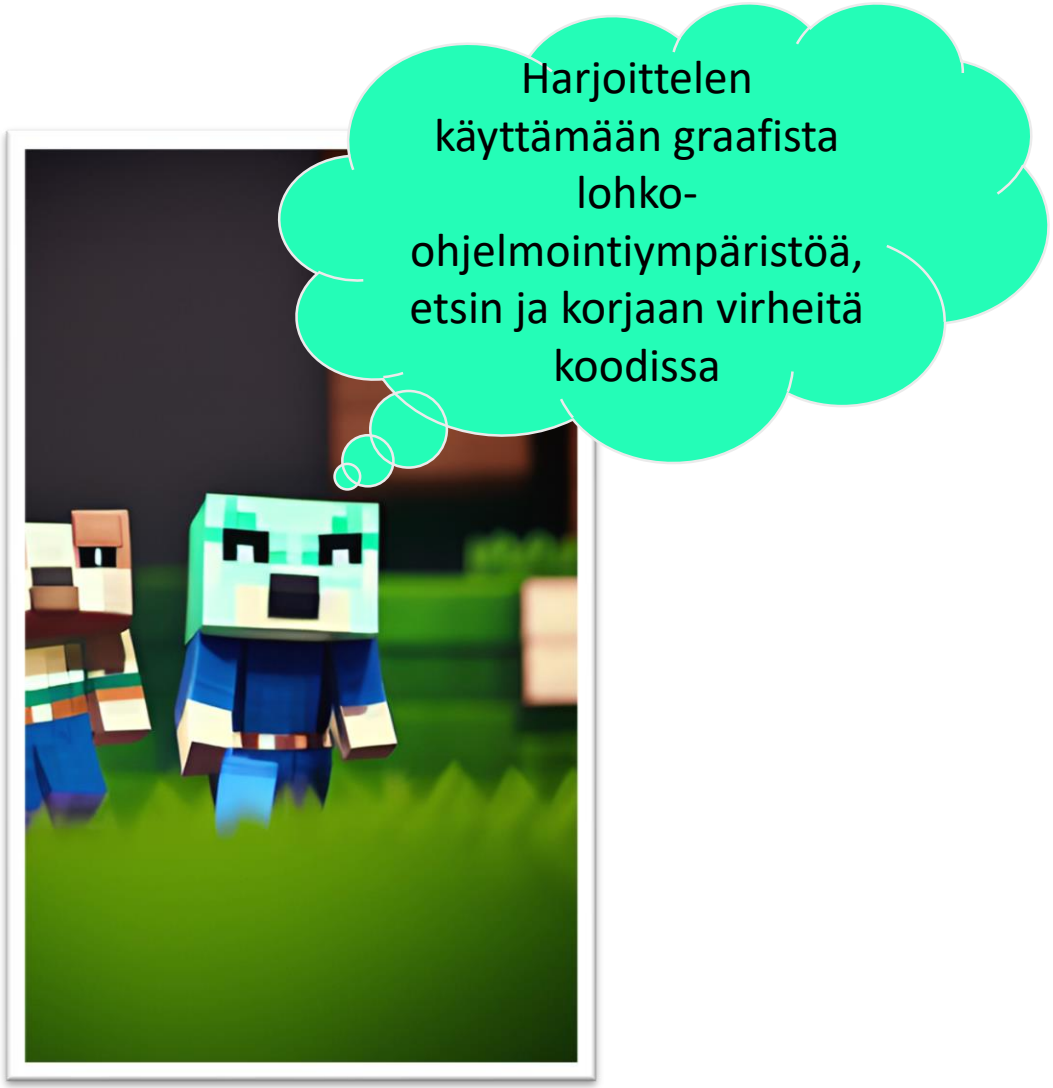
Ohjelmoitujen ympäristöjen tunteminen ja niissä toimiminen

Oppilas
- havaitsee ohjelmoinnin ja robotiikan läsnäolon ympäröivässä yhteisössä

# Code.org

Code.org on yhdysvaltalaisen voittoa tavoittelemattoman säätiön luoma sivusto, jonka tavoitteena on mahdollistaa tietojenkäsittelytieteiden ja ohjelmoinnillisen ajattelun taitojen oppiminen hauskalla ja helpolla tavalla. Sivusto on saatavilla myös osittain suomeksi.

Hyviä harjoituksia lohko-ohjelmointiin tutustumiseen ovat mm. Minecraft Seikkailija ja Flappy Code. Muistakaa katsoa myös opetusvideot!



# Sphero indi

Ohjelmoinnissa on muutamia käsitteitä, joita käytetään usein, kuten esimerkiksi e*hdot, toistot, silmukat, funktiot* ja *algoritmit.*

Osa näistä on saattanut tulla tutuksi jo alemmilla luokilla tai muissa ohjelmointiympäristöissä.

Käsitteitä voi kuitenkin kerrata halutessaan Indi tutuksi –materiaalin sekä siihen liittyvien tukivideoiden avulla.

Kannattaa myös kannustaa oppilaita omiin luoviin ratkaisuihin!

# Lego Spike

Tutustun robotiikkaan, havainnoin ohjelmoinnin ja robotiikan läsnäoloa ympäröivässä yhteiskunnassa, harjoittelen käyttämään graafista lohko-ohjelmointiympäristöä

Legon uusimmat ohjelmoitavat robotit hyödyntävät Scratch-pohjaista ohjelmointikieltä. Tämä on erinomainen asia, koska tästä syystä työskentely Spikella ja Scratchissa on luontevaa, kun oppii kielen toisessa.

Tutustumisen Spike-robottiohjelmointiin voi aloittaa 3. – 4. luokilla siihen tehdyn materiaalin avulla tai vaihtoehtoisesti Legon sivuilta kohdasta **Aloita**.

Hyvä tavoite tässä vaiheessa on saada toteutettua yksinkertaisia ohjelmia pelkällä Spike hubilla (esim. hymiöt) tai vaihtoehtoisesti rakentaa yksinkertainen liikkuva robotti.

# Scratch ja Scratch Jr.

Scratch Jr. käyttöliittymän pitäisi olla oppilaille tuttu jo aiemmilta luokilta, mutta mikäli näin ei ole, voi oppilaiden kanssa käydä läpi Auroran Scratch Jr. ohjelmointikoulun.

Scratch on Michiganin teknillisen yliopiston luoma koulutuksessa yleiskäytössä oleva lohko-ohjelmointiympäristö. Sen käyttöönotto vaatii hieman perehtymistä, mutta on hyödyllistä, koska se on kaikkein laajimmin käytössä oleva lohko-ohjelmointikieli. Apuna voi käyttää mm. Pulmaarion Ohjaajan opasta.

# Esimerkkejä vuoden aikana toteutettavista ohjelmointiprojekteista

## 3. Luokalla (Paljon kertausta)

- Sphero Indi tutuksi
  - Tehtävät 1-4
- Auroran Scratch Junior –ohjelmointikoulu johonkin oppiaineeseen eheytettynä
- Tutustumista lohko-ohjelmointiin code.org:ssa
  - Minecraft-seikkailija
  - Flappy-bird

## 4. luokalla

- Lisää pelillistettyjä lohko-ohjelmointiharjoituksia esim. code.org:ssa
  - Minecraft: Merellinen matka
- Tutustumista koordinaatistoon ja lohko-ohjelmointia Scratchilla
  - Pulmaario: Meriseikkailu (s. 37-)
- Internet of Things ja robotiikka, tutustumista Lego Spike –robottiin
  - Lego Spike -ohjelmointi

# Lisäksi

- Tutustuminen omaan digitaaliseen jalanjälkeen, evästeisiin sekä verkon kohdennetun sisällön algoritmeihin
- Matemaattisen tiedon, lukujen ja mittayksiköiden hahmottaminen erilaisten apuvälineiden ja symbolien avulla
- Lisää koodattavaa: Koodimatskut.fi

# Vuosiluokat 5 - 6

Oppilas hallitsee ohjelmointitaitoja

# Yleistavoitteet

Ohjelmoinnillinen ajattelu

Oppilas
- hahmottaa kokonaisuuksia, löytää ja kuvaa eri ilmiöiden välisiä yhteyksiä sekä syyn ja seurauksen välisiä suhteita

Tutkiva työskentely ja tuottaminen

Oppilas
- osaa tehdä yhteistyötä ja työskennellä kestävästi yhteisen tavoitteen saavuttamiseksi
- jalostaa olemassa olevia ratkaisuja ja harjoittelee toistuvaa työtä eli ideoiden kehittämistä, valmistamista, testaamista ja edelleen kehittämistä useaan otteeseen

Ohjelmoitujen ympäristöjen tunteminen ja niissä toimiminen

Oppilas
- osaa antaa esimerkkejä kohdistetusta digitaalisesta sisällöstä ja sen kohdistamisesta
- pohtii omaa toimintaansa digitaalisissa ympäristöissä ja sitä, mitä tietoa hän itse jakaa Internetissä

# Erityistavoitteet

Ohjelmoinnillinen ajattelu

Oppilas
- arvioi ratkaisuja käyttämällä kriteerejä, kuten toimivuutta, luettavuutta tai tehokkuutta osaa etsiä ja korjata ohjelmassa virheitä.
- osaa käyttää toisto- ja ehtolauseita graafisessa ohjelmointiympäristössä robotin ohjaamiseen eri antureilla
- osaa käyttää graafisia ohjelmointiympäristöjä ja tutustuu robotiikkaan

Tutkiva työskentely ja tuottaminen

Oppilas
- osaa käyttää tuotteissaan omia havaintojaan, mittauksiaan ja antureitaan ja yhdistää niitä automaatioon ja robotiikkaan
- käyttää ohjelmointiin liittyviä työmenetelmiä ja työkaluja tutkiakseen ja selittääkseen erilaisia ilmiöitä eri aineissa ja monitieteisissä oppimisprojekteissa

Ohjelmoitujen ympäristöjen tunteminen ja niissä toimiminen

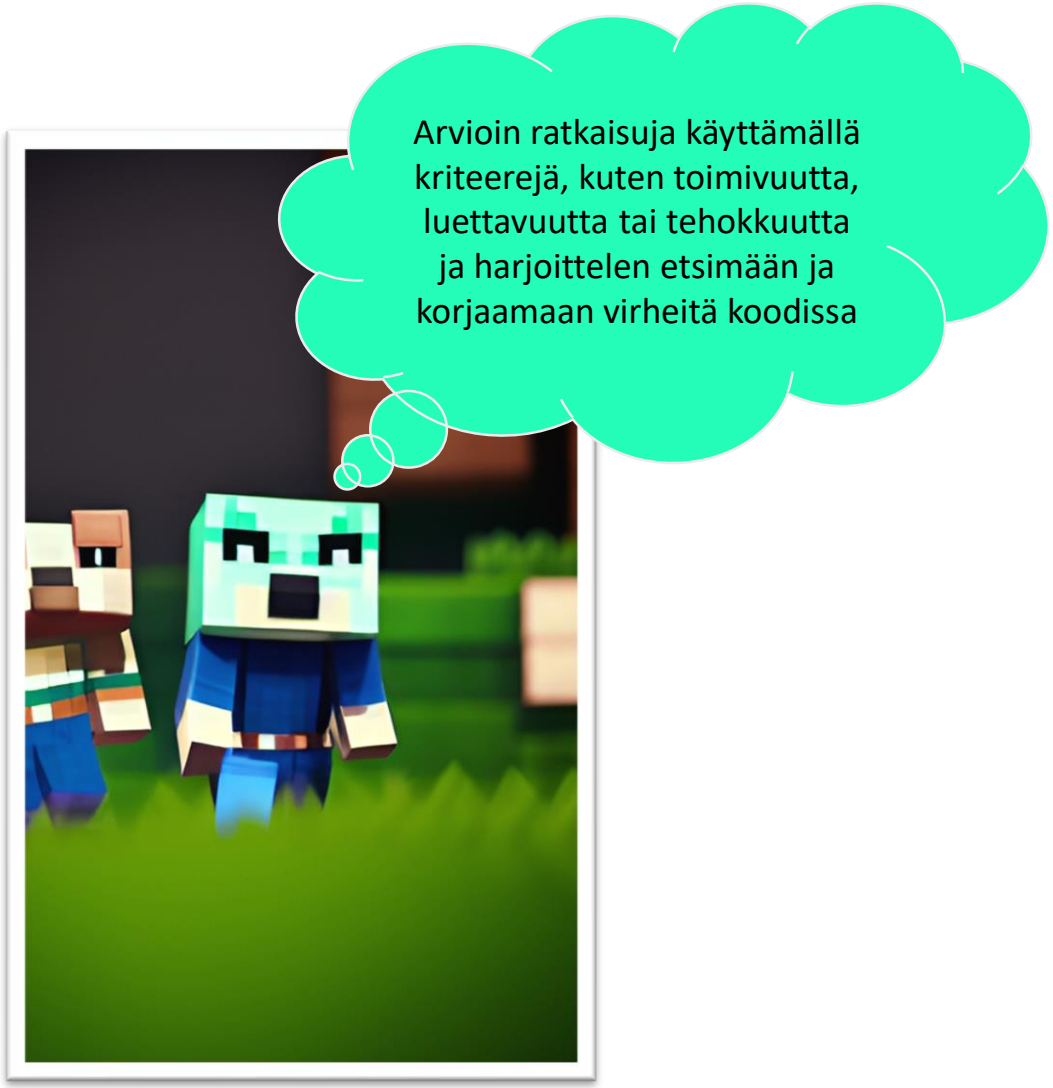
Oppilas
- osaa kertoa erilaisista teknisten sovellusten käyttötavoista ja niiden toimintaperiaatteista sekä kuvailla niiden merkitystä omassa elämässään

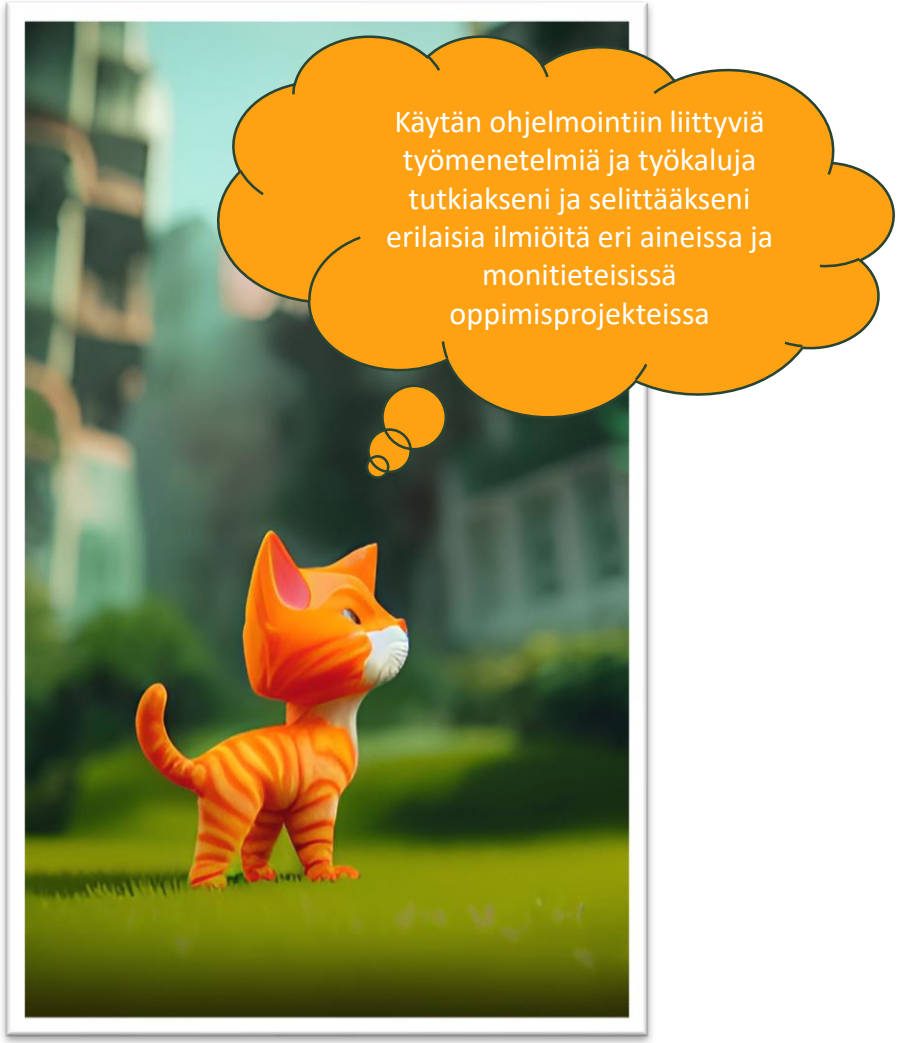
# Code.org

Ohjelmoinnissa on monta tapaa päästä tavoitteen mukaiseen lopputulokseen. Oppilaita onkin hyvä kannustaa siihen, että yhtä oikeaa ratkaisua ei ole ja keskeistä on ongelman ratkaiseminen jollain tavalla.

On kuitenkin myös tapoja arvioida koodin laatua. Esimerkiksi toisto- ja ehtolausekkeilla saadaan koodista usein huomattavasti luettavampaa ja helpompaa muokata kuin ns. kovakoodaamalla.

Ohjelmoinnillisen ajattelun kehittyessä on hyvä palata vanhoihin harjoituksiin code.org:ssa ja kokeilla, ovatko ratkaisut kehittyneet ja parantuneet.






# Scratch

Scratch on toivottavasti tullut oppilaille tutuksi jo aiemmilla vuosiluokilla. Se on lohko-ohjelmointikieli, joka toimii selaimessa olevan sivuston kautta.

Scratchilla pystyy toteuttamaan monenlaisia pieniä pelejä ja animaatioita, joiden sisältöjä pystyy tuomaan eri oppiaineista.

Apuna voi käyttää mm. Pulmaarion Ohjaajan opasta.

# Lego Spike

Useimmat tekniset laitteet, kuten jääkaapit, autot, kuumemittarit ja jopa lemmikkien ruokinta-automaatit ovat tietokoneohjattuja. Niiden aivoina toimii pienoistietokone eli mikrokontrolleri.

Myös Legon Spike –robotin aivot eli Spike Hub on tällainen ja siksi sillä pystyy toteuttamaan vaikka minkälaisia keksintöjä.

Käytän toisto- ja ehtolauseita graafisessa ohjelmointiympäristössä robotin ohjaamiseen eri antureilla

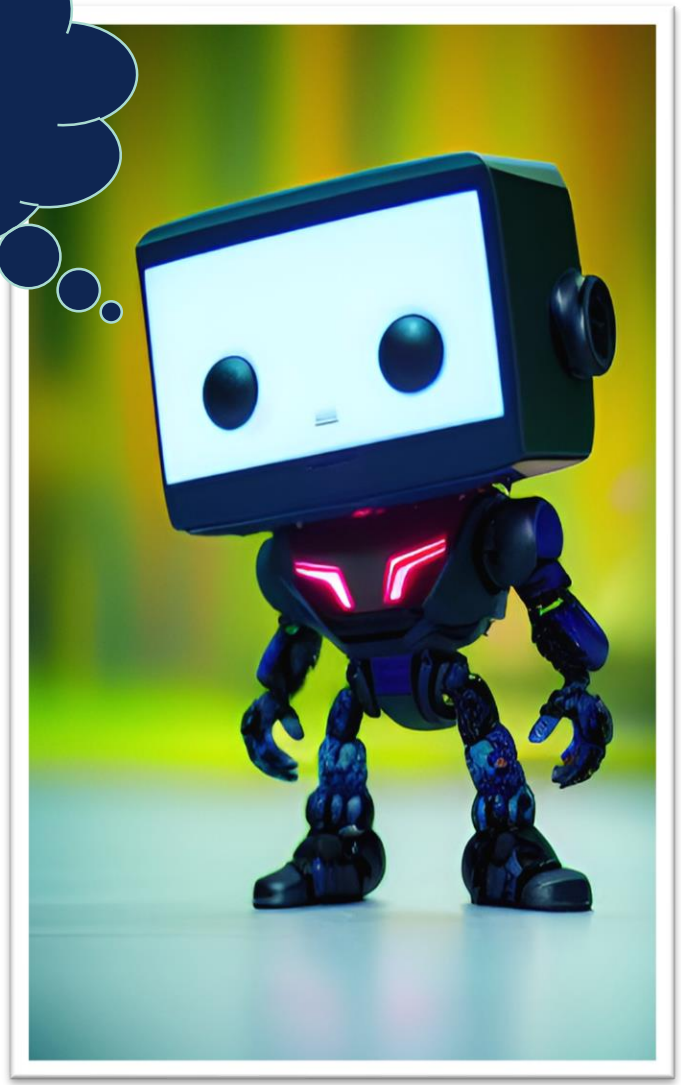
# Micro:bit 1/2

Siinä missä Lego ja Scratch keskustelevat hyvin keskenään, keskustelevat myös Micro:bitin käyttämä Makecode ja Code.org. Niiden käyttämä lohko-ohjelmointikieli on keskenään samanlaista ja lisäksi hyvin lähellä Scratchia.

Micro:bit on myös mikrokontrolleri eli tietokone pienoiskoossa. Sillä pystytään mittaamaan monenlaisia asioita (kiihtyvyys, valoisuus, melutaso, lämpötila ym.) ja liittämään koodia näiden tulostamiseen näytölle tai saadun datan työstämiseen.

# Micro:bit 2/2

Osaan kertoa erilaisista teknisten sovellusten käyttötavoista ja niiden toimintaperiaatteista sekä kuvailla niiden merkitystä omassa elämässäni



Micro:bitille ja sen lisäosille löytyy paljon valmista materiaalia. Ohjelmoinnin voi aloittaa esimerkiksi Innokas-verkoston materiaalipankin perusteista ja ohjelmoinnin rakenteista.

Valmiita jaksoja ja myös opettajan ammatillista kehittymistä tukevia oppimispolkuja löytyy täältä: Professional Development courses.

# Esimerkkejä vuoden aikana toteutettavista ohjelmointiprojekteista

## 5. luokalla

- Scratch-animaatio tai pieni peli esimerkiksi osana historian jaksoa tai MOKia
  - pohjana voi käyttää Auroran Scratch-animaatiokoulua tai pelissä Pulmaarion ohjaajan opasta
- Arkea hyödyttävä keksintö Lego Spikella
  - Ideoita ja malleja voi katsoa Legon Spike-sivustolta kohdasta Jaksot
- Harjoituksia code.org-sivustolta, ajatuksena ohjelmoinnillisen ajattelun kehittyminen ja aiemmin keksittyjen ratkaisujen parantelu

## 6. luokalla

- Tekoäly (Artificial Intelligence) ja koneoppiminen integroituna englantiin
  - Code.org/ai
- Lego Spike –turnausrobotin rakentaminen ja ohjelmointi
  - Tutustu lajeihin
- Tutustuminen Micro:bitiin
  - Perusteet
  - Ohjelmoinnin rakenteet
  - Erilaisia projekteja ja opetusmateriaaleja
  - Tervetuloa työhaastatteluun! -tehtävät

# Lisäksi

- Harjoitellaan systemaattisesti ja monessa eri yhteydessä hahmottamaan kokonaisuuksia, löytämään ja kuvaamaan eri ilmiöiden välisiä yhteyksiä sekä syiden ja seurauksien välisiä suhteita
- Harjoitellaan yhteistyötaitoja ja työskentelyä pitkäjänteisesti yhteisen tavoitteen saavuttamiseksi
- Opetellaan iteratiivista työtapaa eli jalostamaan olemassa olevia ratkaisuja sekä harjoitellaan ideoiden kehittämistä, valmistamista, testaamista ja edelleen kehittämistä useaan otteeseen
- Lisää koodattavaa: Koodimatskut.fi