

Polkuja

ohjelmointiosaamiseen

OPAS
VUOSILUOKILLE
7-9



Polkuja ohjelmointiosaamiseen -oppaan vuosiluokille 7–9 on tuottanut Kansallinen audiovisuaalinen instituutti KAVI. Opasta on ollut laatimassa yhteistyössä KAVIn kanssa ohjelmoinnin opetuksen ja mediakasvatuksen asiantuntijoita Suomen Koodikoulusta, kehittämisohjelman työryhmistä sekä perusopetuksen kentältä. Kiitokset kaikille oppaan tekemisessä mukana olleille!

Opas on julkaistu osana valtakunnallista Uudet lukutaidot -kehittämisohjelmaa (2020–2023). Ohjelma tarjoaa välineitä medialukutaidon, ohjelmointiosaamisen sekä digitaalisen osaamisen ja tieto- ja viestintäteknologisen osaamisen edistämiseen varhaiskasvatuksessa sekä esi- ja perusopetuksessa. Ohjelman on rahoittanut opetus- ja kulttuuriministeriö sekä toteuttanut Kansallinen audiovisuaalinen instituutti KAVI ja Opetushallitus.

Opas kuuluu Polkuja ohjelmointiosaamiseen -sarjaan, mistä löydät oppaat myös varhaiskasvatukseen ja esiopetukseen sekä vuosiluokille 1–6. Vastaava oppasarja on julkaistu myös medialukutaitoon.

Lisätietoa ja oppaat löydät www.uudetlukutaidot.fi

2. uudistettu painos

Tekstit: Jussi Koivisto ja Tarmo Toikkanen, Suomen Koodikoulu Oy

Lauri Palsa, Kansallinen audiovisuaalinen instituutti

Kuvitus: Emma Kantanen

Taitto: Kiira Koivunen ja Tuomas Korolainen, Mene Creative Oy

ISBN 978-952-7475-21-8 (.nid),

Kansallisen audiovisuaalisen instituutin julkaisuja 9/2022 (ISSN 2342-5970)

ISBN 978-952-7475-22-5 (PDF),

Kansallisen audiovisuaalisen instituutin julkaisuja 9/2022 (ISSN 2342-5970)

[Creative commons -lisensointi: Nimeä 4.0 Kansainvälinen \(CC BY 4.0\)](https://creativecommons.org/licenses/by/4.0/)



KANSALLINEN AUDIOVISUAALINEN INSTITUUTTI
NATIONELLA AUDIOVISUELLA INSTITUUTET
NATIONAL AUDIOVISUAL INSTITUTE



UUDET
LUKUTAI-
DOT

Kansallinen audiovisuaalinen instituutti on mediakasvatusviranomainen, joka lain (1434/2007) mukaisesti edistää mediakasvatusta, lasten mediataitoja ja lapsille turvallista mediaympäristöä. Osallistumme aktiivisesti yhteiskunnalliseen keskusteluun mediakasvatuksesta sekä tiedotamme lapsiin ja mediaan liittyvistä asioista. Meihin voi olla aina yhteydessä mediakasvatusta ja medialukutaitoa koskevissa asioissa.

Sisällys

Avuksi ohjelmointiosaamisen poluille vuosiluokilla 7-9.....	4
Näin käytät opasta	5
1 Ohjelmoinnillinen ajattelu	6
Hyvä osaaminen vuosiluokilla 7-9.....	8
1. Yleistetään ilmiöitä kaavoilla	9
2. Suunnitellaan algoritmeja	10
3. Python-ohjelmointikielellä alkuun.....	10
4. Piirretään ohjelmoimalla	11
5. Rakennetaan reaktiopeli	12
6. Tutkitaan ja tuotetaan musiikkikappaleita.....	13
7. Tehdään tietovisa.....	14
8. Tietoturva-asiat kuntoon salasanageneraattorilla	14
9. Käsitellään eri tietotyyppäjä.....	15
10. Itsenäistä harjoittelua verkossa	16
11. Avoimia oppimateriaaleja ohjelmointiin	17
2 Tutkiva työskentely ja tuottaminen	18
Hyvä osaaminen vuosiluokilla 7-9.....	20
12. Tehdään lautapelejä.....	21
13. Herätetään maalaukset eloon	22
14. Tehdään yhdessä pelejä	23
15. Kilpaillaan roboteilla	24
16. Automatisoidaan Minecraftissa.....	25
17. Ohjelmoidaan kielentunnistusalgoritmi	26
18. Tehdään interaktiivisia esitelmiä ilmiöistä	27
19. Ohjelmoidaan esineitä tunnistava algoritmi	28
20. Mallinnetaan ilmiöitä 3D-ympäristössä.....	29
21. Luodaan arjen automaatiota	30
22. Rakennetaan pakuhuone.....	31
3 Ohjelmoidut ympäristöt ja niissä toimiminen	32
Hyvä osaaminen vuosiluokilla 7-9.....	34
23. Haetaan tietoa ja opitaan hakemaan	35
24. Kaikelle on syy	36
25. Kone vastaan ihminen	37
26. Sulautetut järjestelmät	38
27. Robotiikan kolme sääntöä	39
28. Video juuri sinulle!	40
29. Luokan tietokannat	41
30. Tarvitaanko näitä tietoja oikeasti?	41
Uudet lukutaidot tukena digiajan opetuksessa ja kasvatuksessa	42

Avuksi ohjelmointiosaamisen poluille vuosiluokilla 7–9

Tämä opas tarjoaa sinulle pedagogisia reittejä ja käytännönläheisiä esimerkkejä ohjelmointiosaamisen edistämiseen yläkouluikäisten nuorten opetuksessa.

Ohjelmointiosaaminen on kaikille tarpeellinen kansalaistaito. Kyse ei ole pelkästään koodaamisesta, vaan osaamiseen kuuluvat olennaisesti myös monipuoliset ajattelun ja yhteistyön taidot sekä ymmärrys digitaalisesta, ohjelmoidusta maailmasta ja siinä toimimisesta sekä siitä, mitä kaikkea ohjelmoimalla voi saada aikaan. Kaikilla oppilailla on oikeus saada yhdenvertaisesti opetusta ohjelmointiin sekä hyvään elämään ja aktiiviseen toimintaan ohjelmoiduissa ympäristöissä.

Nuorilla on monenlaisia mediankäytön taitoja ja digitaaliset ympäristöt tarjoavat heille työkaluja ja kanavia oman äänen esille tuomiseen – itseilmaisuun, osallistumiseen ja vaikuttamiseen. Osaaminen ja kokemukset ohjelmoinnista voivat kuitenkin vaihdella esimerkiksi johtuen nuoren omasta kiinnostuksesta tai aiemmasta kouluopetuksesta. Yläkoulussa on tarpeen vahvistaa oppilaiden osaamista ja valmiuksia toimia itsenäisesti ja yhdessä ohjelmoiduissa ympäristöissä sekä käyttää ohjelmointia luovaan tekemiseen, itseilmaisuun ja oikean elämän kysymysten ja ongelmien ratkomiseen.

Jokaisella opettajalla on mahdollisuus edistää oppilaiden ohjelmointiosaamista, ja tavoitteisiin voi päästä suhteellisen yksinkertaisellakin tekemisellä. Oppiaineesta riippumatta oppilaiden kanssa voidaan pohtia ja tutkia, miten ohjelmoitu teknologia, erilaiset digitaaliset ympäristöt ja ohjelmointi linkittyvät esimerkiksi automaation, simulaation, datan tai pelien kautta kyseiseen tiedonalaan ja sen oppimiseen. Ohjelmoinnillista ajattelua ja yhteiskehittelyn työtapoja voidaan harjoitella monien oppiaineiden ja -sisältöjen käsittelyssä ja niihin liittyvässä omassa tuottamisessa, erityisesti oppilaita innostavassa itseilmaisussa ja luovissa kokeiluissa.

Vuonna 2021 ensimmäisen kerran julkaistuun oppaaseen on nyt päivitetty Uudet lukutaidot -kehittämishjelmassa vuonna 2022 uudistetut *oppilaan hyvän ohjelmointiosaamisen kuvaukset* vuosiluokille 7–9. Hyvän vastaanoton saaneet käytännön esimerkit ovat oppaassa ennallaan ja ne soveltuvat edelleen päivitettyjen kuvausten toteuttamiseen. Kuvaukset ja pedagogiset esimerkit pohjautuvat perusopetuksen opetussuunnitelman perusteisiin (2014), ja niiden tarkoituksena on tukea ja kannustaa monipuoliseen ohjelmoinnin opetukseen osana eri oppiaineita. Ohjelmointiosaaminen linkittyy opetussuunnitelmassa vahvasti laaja-alaiseen osaamiseen: erityisesti monilukutaitoon sekä tieto- ja viestintäteknologiseen osaamiseen.

Toivomme, että opas innostaa sinua yhdessä työyhteisön kanssa hyödyntämään osaamisen kuvauksia ja rohkaisee nivomaan ohjelmointiosaamisen teemoja opetukseen.

Näin käytät opasta

Ohjelmointiosaaminen jakaantuu kolmeen pääalueeseen: *ohjelmoinnilliseen ajatteluun, tutkivaan työskentelyyn ja tuottamiseen sekä ohjelmoituihin ympäristöihin ja niissä toimimiseen*. Kunkin pääalueen sisällä avataan tarkemmin opetuksen tavoitteita ja sisältöjä sekä sitä, miten asioita voi käsitellä oppilaiden kanssa monipuolisesti. Käytännön työskentelyssä alueiden sisällöt linkittyvät toisiinsa.

Oppaan esimerkit on tarkoitettu opetuksen suunnitteluun ja toteutukseen. Voit hyödyntää ideoita sellaisenaan tai soveltaa niitä omalle oppilasryhmällesi sopivaksi. Opas sisältää myös koontina aiemmin tuotettuja avoimia materiaaleja.

Esimerkeissä ja niiden järjestyksessä on huomioitu vuosiluokkakokonaisuuksien tavoitteet ja oppilaiden taitojen kehittyminen. Työskentelyn vaativuus on esitetty symbolein  (1–3). Jos ohjelmoinnin teemat ovat itselle tai oppilaille vielä uusia, kannattaa aloittaa yhden symbolimerkin alta löytyvistä helpoimmista esimerkeistä.

Opettajan tehtävänä on luotsata oppilaitaan ohjelmoinnilliseen ajatteluun, kokeilemaan ohjelmointia sekä pohtimaan ohjelmoituja ympäristöjä ja omaa toimintaansa niissä. Ohjelmoinnin teemojen nivominen opetukseen saattaa tuntua ensisilmäyksellä haastavalta. Kaikkea ei kuitenkaan tarvitse tehdä ja osata heti; opettajakin saa olla oppijan roolissa ja ottaa asioita haltuunsa vähitellen. Oppaan esimerkkien avulla sekä oppilaiden kanssa yhdessä kokeilemalla ja tekemällä jokainen voi löytää omat polkunsä ohjelmoinnin maailmaan.

Innostavia oivalluksia ja tekemisen iloa ohjelmointiosaamisen poluille!

1

Ohjelmoinnillinen ajattelu



Ohjelmointiosaamista edistetään kehittämällä erityisesti oppilaiden ohjelmoinnillista ajattelua. Opetuksessa painottuvat loogista ajattelua, monipuolista tiedon käsittelyä ja ongelmanratkaisua kehittävä työskentely sekä ohjelmoinnin käsitteistöä ja ohjelmointia hyödyntävät kokeilut ja tekeminen. Oppilaille tarjotaan mahdollisuuksia tutkia ja tehdä valintoja, etsiä ja korjata virheitä sekä ratkoa oikeita ongelmia eri oppisisältöihin ja ympäröivän maailman ilmiöihin liittyen.

Yläkoulussa ohjelmoinnillisen ajattelun taitoja syvennetään. Tietoa opetellaan hankkimaan, käsittelemään ja esittämään erilaisten yleistysten, kaavioiden, merkin-tätapojen ja toimien avulla. Oppilaita ohjataan selvittämään monenlaisia haasteita: analysoimaan ja arvioimaan ongelmia ja niiden mahdollisia ratkaisuja, suunnitte-lemaan ja kokeilemaan erilaisia ratkaisumalleja sekä visualisoimaan ongelmia ja niiden ratkaisuja.

Ohjelmointi sopii tiedon käsittelyyn ja ongelmanratkaisuun kaikissa oppiaineissa. Ohjelmoitaessa laaditaan algoritmeja eli yksityiskohtaisia ohjeita toiminnasta, jolla jokin ongelma saadaan ratkottua tai tehtävä suoritettua. Yläkoulussa algoritmeja suunnitellaan ja laaditaan käyttäen apuna kaavioita, joiden pohjalta voidaan tuottaa myös koodia ja ohjelmia.

Ohjelmoinnillista ajattelua ja käytännön ohjelmointitaitoja harjoitellaan yläkoulussa käyttämällä tarkoituksenmukaisesti erilaisia työskentelytapoja, ympäristöjä ja välinei-tä. Toisten tuottamat koodit ja valmiit ohjelmat ovat keskeinen käytettävä materiaali. Koodia opetellaan lukemaan ja tulkitsemaan. Tutkimalla ja hyödyntämällä olemassa olevia koodeja omien tuotosten tekemisessä, opitaan myös ymmärtämään, miten koodi toimii. Kun oppilaat tuntevat ohjelmoinnin peruskäsitteet, on luontevaa ja mielekäästä tutustua graafisten ohjelmointiympäristöjen lisäksi myös tekstipohjaiseen ohjelmointiin.

Hyvä osaaminen vuosiluokilla 7–9

Looginen ajattelu ja tiedon käsittely

Oppilas osaa käsitellä erilaisia informaation muotoja, käyttää sen käsittelyssä erilaisia merkitäytöjä ja toteuttaa informaation kanssa loogisia operaatioita.

Oppilas hyödyntää algoritmista ajattelua eri oppiaineisiin ja projekteihin liittyvässä tutkimisessa sekä tiedon tuottamisessa ja esittämisessä.

Ongelmien ratkaiseminen ja mallintaminen

Oppilas osaa analysoida ongelmia ja arvioida niiden mahdollisia ratkaisuja erilaisten kriteerien perusteella sekä visualisoi ongelmia ja ratkaisuja yleistysten ja kaavioiden avulla.

Oppilas hyödyntää algoritmista ajattelua eri oppiaineisiin ja projekteihin liittyvässä ongelmanratkaisussa.

Ohjelmointiin liittyvä toiminta, käsitteet ja perusrakenteet

Oppilas tietää, miten erilaiset ohjelmointirakenteet, kuten peräkkäiset, toistuvat ja ehdolliset toiminnot sekä muuttujat, toimivat. Oppilas osaa suunnitella ohjelman niitä käyttäen.

Oppilas osaa tulkita tekstipohjaista ohjelmointikieltä ja tunnistaa erilaisia rakenteita sillä tehdystä ohjelmakoodista.





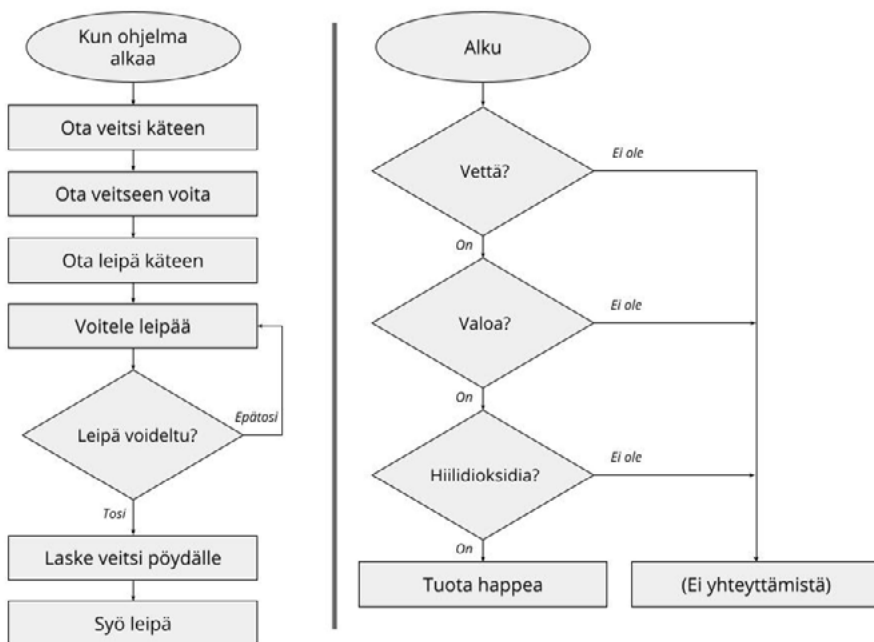
1. Yleistetään ilmiöitä kaavoilla

Opitaan ongelmien ja ilmiöiden visualisointia ja yleistämistä kaavioiden avulla. Tehdään yleistyksiä monimuotoisista ilmiöistä. Opitaan peräkkäisyyden, toistuvuuden ja ehdollisten toimintojen käyttämistä.

Kuvatkaa oppiaineen kannalta ajankohtaisia ilmiöitä tai toimintoja vuokaavioina, tehden yleistyksiä. Voitte ensin harjoitella kaavioiden laatimista arjen ongelmia ja toimenpiteitä mallintamalla. Erottaa erityyppiset vaiheet, kuten toimenpiteet ja ehdot (tosi / epätosi) toisistaan eri muotoisilla laati-koilla. Voitte lopuksi antaa yhdelle oppilaalle tehtäväksi seurata mallia robottimaisesti, jotta näette, toimiiko se oikein.

Vinkki!

Jo yksinkertainen prosessi paisuu helposti varsin suureksi, mutta tavoite on tehdä yleistyksiä. Monisyisiä ilmiöitä (yhteyttäminen, ukkonen, kemialliset reaktiot, auton moottorin toiminta tai ilon tunne) mallintaessa joudutaan tekemään vahvoja yleistyksiä, jolloin poikkeustilanteet täytyy usein jättää pois.



Kaavioiden piirtoa ja logiikkaa voidaan harjoitella arjen askareiden mallintamisella. Käytetyt symbolit ovat vuokaavioissa yleisesti käytettyjä. Selitykset löydät mm. englanninkielisestä Wikipediasta hakusanalla "[flowchart](#)".



Vuokaavioita käytetään usein tietokoneiden suorittamien algoritmien suunnittelussa. Monimuotoisten ilmiöiden kohdalla tulee pohtia, mitkä ovat niitä poikkeuksia, joissa tehty yleistys ei enää päde. Kaavioiden tekemisessä saattaa auttaa, kun ajatellaan tekijän olevan robotti, esimerkiksi robottikasvi, joka yhteyttää.



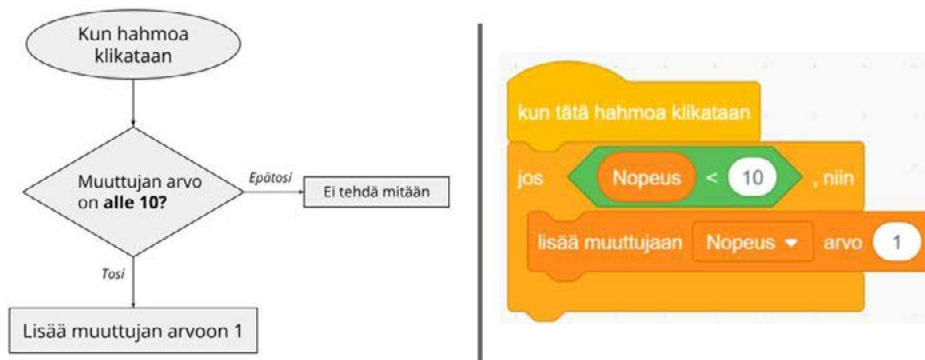
2. Suunnitellaan algoritmeja

Opitaan ohjelmakoodin suunnittelua kaavioiden avulla. Opitaan peräkkäisyyden, toistuvuuden ja ehdollisten toimintojen käyttämistä.

Ohjelmoinnissa on tärkeää etukäteissuunnittelu. Käyttäkää ohjelmoinnissa suunnittelun työkaluna vuokaavioita. Erottaa erityyppiset vaiheet, kuten toimenpiteet ja ehdot (tosi / epätosi) toisistaan eri muotoisilla laatikoilla (esimerkki alla). Eri ryhmät voivat katselmoida toistensa vuokaaviot, jolloin osa virheistä löytyy jo ennen ohjelmoinnin aloittamista.

Vinkki!

Katso myös edeltävä **Yleistetään ilmiötä kaavoilla** -esimerkki. Käytetyt symbolit ovat vuokaavioissa yleisesti käytettyjä. Selitykset löydät mm. englanninkielisestä Wikipediasta hakusanalla "[flowchart](#)".



Tässä kuvaus, miten kaaviosta johdetaan "scratch"-koodia.



3. Python-ohjelmointikielellä alkuun

Opitaan perusteita ohjelmointikielestä. Ohjelmointiosaaminen on laajempi aihe kuin varsinainen ohjelmointi ja koodaus. Ohjelmoinnissa voi kuitenkin lähteä liikkeelle tutustumalla erilaisiin ohjelmointikieliin ja niiden toimintaan. Tässä esimerkissä on koottu kaksi eri tapaa tutustua Python-ohjelmointikielen perusteisiin.

Perehtykää Pythonin perusteisiin testaten ja tutkien valmiin neljäosaisen harjoituspaketin avulla (suositellaan tulostettavaksi). Harjoituksissa on tarkoitus kirjoittaa ohjeen mukaista koodia ja ratkaista haastetehtäviä. Paketissa korostuu koodin kommentointi sekä eri tietotyypit ja niiden keskinäinen käsittely. Tähän on hyvä tutustua huolella, sillä juuri tietotyypit ovat osoittautuneet yhdeksi aloittelevan ohjelmoijan kompastuskivistä. Löydät tulosteet [linkin kautta](#).

Verkkopalveluun on rakennettu polku Pythonin alkeiden itseopiskeluun. Palvelun käyttö vaatii tunnuksen luomista, mutta ei edellytä tunnistettavien henkilötietojen syöttämistä. Löydät kurssin [linkin takaa](#).



4. Piirretään ohjelmalla

Opitaan tekstipohjaisen ohjelmointikielen käyttämistä sekä siihen liittyvää koodin korjaamista ja ongelmanratkaisua. Tuotetaan omia luovia tuotoksia kokeilun kautta. Opitaan peräkkäisyyden ja toistuvuuden käyttämistä.

- ◆ Ohjelmointia varten tarvitsette ohjelmointiympäristön. Tässä esimerkissä käytetään Pythonin Turtle-moduulia.

Tutustukaa Pythonin Turtle-moduuliin piirtämällä geometrisia muotoja. Voit tutustua moduuliin [täällä](#). Tutustumisen jälkeen voitte pohtia esimerkiksi, kuinka paljon on käännyttävä, jotta saadaan aikaan kolmio tai viisikulmio? Entä onnistuuko tähden, kukan tai pallon piirtäminen?

Lisätkää kuvioihin toistoa, pieniä muutoksia (esim. kolmen asteen käännös joka piirtokerran välissä), satunnaisuutta sekä värien vaihtelua. Geometria muuttuu visuaaliseksi taiteeksi. Millaisia tähtiä, kukkia tai serpentiinejä nyt saatte aikaan?

Vinkki!

Aloituksen sujuvoittamiseksi voitte käyttää pohjaa [kilpikonnan ohjelmointiin](#).

- ◆ Esimerkin kilpikonnataiteesta löydät [täältä](#).
- ◆ Esimerkin kilpikonnakellosta löydät [täältä](#).



Piirto-ohjelmointi on hyvä ensikoketus tekstuaaliseen ohjelmointiin. Perusohjelmointaitojen lisäksi tässä hahmottuvat x/y-koordinaatisto sekä trigonometrian perusteet.



Kilpikonna-taide voi olla esimerkiksi sarja erivärisiä viivoja mielenkiintoisessa muodossa.



Kilpikonnat ovat ikään kuin hahmoja, jotka voidaan myös korvata omilla kuvilla, jolloin avautuu aivan uusi ulottuvuus kilpikonna-ohjelmointiin. Voit esimerkiksi korvata kilpikonnat piirretyillä viisareilla ja ohjelmoida analogisen kellon.



5. Rakennetaan reaktiopeli

Opitaan ongelman analysointia ja pilkkomista. Opitaan ohjelmakoodin suunnittelua kaavioiden avulla. Ratkaistaan ongelmia aktiivisesti omien havaintojen perusteella. Opitaan peräkkäisyyden, toistuvuuden ja ehdollisten toimintojen käyttämistä.

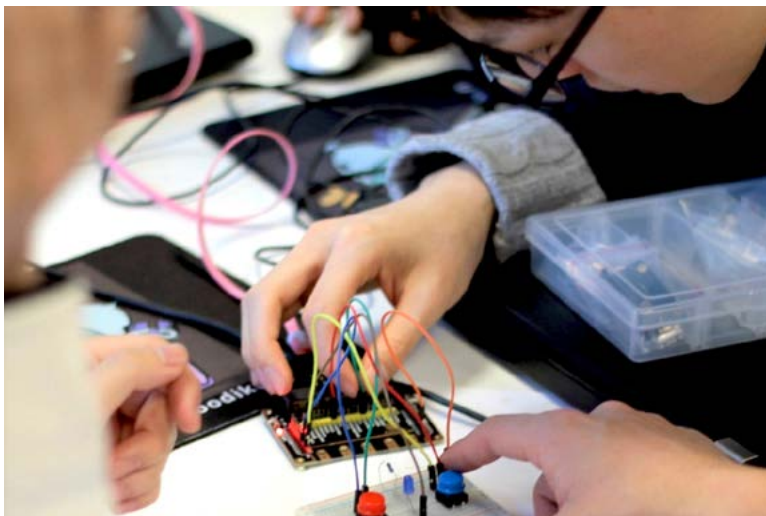
Rakennetaan ja ohjelmoidaan reaktiopeli esimerkiksi Micro:bit- tai Arduino-kehitysalustalle käyttämällä erillisiä elektronisia komponentteja (led-valot, painikkeet ja johtimet). Pelissä siis syttyy valo tai kuuluu ääni, jonka jälkeen pelaajan on painettava nappia mahdollisimman nopeasti. Toimintaa kannattaa suunnitella yhdessä tehtäväksi, esimerkiksi parityönä. Voitte käyttää ohjelman suunnittelussa vuokaavioita (katso **Suunnitellaan algoritmeja** -esimerkki).

Reaktiopelin ohjelman [löydät täältä](#).

Esimerkin reaktiopelistä [löydät täältä](#).



Elektroniikan kytkennät tuovat sähkönn toiminnan konkreettisesti esille ja selventävät signaalien kulkemisen logiikkaa. Jos elektroniikkalaitteistoa ei ole, voi reaktiopelin tehdä myös tietokoneella eri ohjelmointiympäristöissä. Tosin tällöin voidaan menettää osa tehtävän viehätystsestä ja tavoitteista.





6. Tutkitaan ja tuotetaan musiikkikappaleita

Opitaan ongelman analysointia, pilkkomista ja kaavojen tunnistamista. Opitaan ohjelmakoodin kaltaisten kokonaisuuksien suunnittelua omien havaintojen ja kaavojen muodostamisen kautta. Tuotetaan omia luovia tuotoksia kokeilun kautta. Opitaan peräkkäisyyden ja toistuvuuden käyttämistä.

Tehkää parityöskentelynä musiikkiluuppeja eli 4–8 tahdin mittaisia musiikkipätkiä, jotka laitetaan toistumaan, ja rakentakaa erilaisista luupeista kokonaisia kappaleita. Tyypillisesti tehdään kolme erilaista luuppia, joita käyttämällä saadaan rakennettua kappale, jossa on säkeistö (A-osa), kertosäe (B-osa) ja välisoitto (C-osa). Pop-musiikissa näitä osia on tyypillisesti käytetty kaavalla "ABABCBB".

Aloittakaa ottamalla esiin oppilaiden omat havainnot ja kokemukset musiikkiluupeista. Kuunnelkaa esimerkiksi elektronista musiikkia etsien ja nimeten kappaleista toistuvia osioita.

Omien luuppien tuottamiseen voitte käyttää esimerkiksi [Sumotunes-sovellusta](#), Scratchia musiikki-laajennuksella tai muuta musiikin tuottamiseen suunnattua sovellusta.



Musiikkinuottien lukeminen ja kirjoittaminen on hyvin samankaltaista kuin ohjelmointi. Molemissa asiat tapahtuvat järjestyksessä, toistuvasti ja sääntöjen mukaan.





7. Tehdään tietovisa

Opitaan tekstipohjaisen ohjelmointikielen käyttämistä sekä siihen liittyvää koodin korjaamista ja ongelmanratkaisua. Tuotetaan omia ohjelmia oppiaineeseen sopivaan käyttötarkoitukseen. Opitaan peräkkäisyyden ja toistuvuuden käyttämistä. Tietovisan laatimisessa harjoitellaan ohjelmointitaitoja, mutta lisäksi tekijän on hallittava tietovisan aihealue.

Tehkää ohjelmia, jotka kysyvät käyttäjältä oppiaineeseen liittyviä kysymyksiä, joihin on yksi selkeä oikea vastaus. Oppilaat voivat näin tehdä esimerkiksi sanakokeita toisillensa kokeisiin harjoittelua varten. Ohjelman täytyy siis kysyä käyttäjältä kysymyksiä ja verrata käyttäjän antamaa vastausta oikeaan vastaukseen. Työskentelyssä kannattaa hyödyntää yhteistä tekemistä, esimerkiksi parityötä. Ohjelman suunnittelussa voi käyttää vuokaavioita (katso **Suunnitellaan algoritmeja** -esimerkki).

Pythonilla toteutetun esimerkkiohjelman [löydät täältä](#)

Scratch-toteutusta varten löydät [pohjan täältä](#).

Vinkki!

Ohjelman toiminta perustuu vahvasti ehtolauseisiin. Mikäli yksittäiseen kysymykseen voidaan hyväksyä useita vastauksia, voidaan käyttää OR-operaattoria (tai).



8. Tietoturva-asiat kuntoon salasanageraattorilla

Opitaan tekstipohjaisen ohjelmointikielen käyttämistä sekä siihen liittyvää koodin korjaamista ja ongelmanratkaisua. Tuotetaan ohjelma, joka ratkaisee oikean elämän ongelman. Ratkaistaan ongelmia aktiivisesti omien havaintojen perusteella. Opitaan peräkkäisyyden, toistuvuuden ja ehdollisten toimintojen käyttämistä.

Keskustelkaa yhdessä ja kootkaa yhteen hyvän salasanan piirteitä. Kokeilkaa omia ja erilaisia keksittyjä salasanoja verkossa luotettavalla [salasana-testerillä](#). Työskentelyssä kannattaa hyödyntää yhteistä tekemistä, esimerkiksi parityötä. Voitte käyttää ohjelman suunnittelussa vuokaavioita (katso **Suunnitellaan algoritmeja** -esimerkki).

Suunnitelkaa ja toteuttakaa ohjelmia, jotka generoivat mahdollisimman hyviä 12-merkin salasanoja. Esimerkin Pythonin salasanageraattorista [löydät täältä](#).



9. Käsittellään eri tietotyyppejä

Opitaan ongelman analysointia ja pilkkomista. Opitaan tekstipohjaisen ohjelmointikielen käyttämistä sekä siihen liittyvää koodin korjaamista ja ongelmanratkaisua.

Tutustukaa tietotyyppien käsittelyyn tekstuaalisella ohjelmointikielellä, esimerkiksi Pythonilla, tehden pieniä ongelmanratkaisutehtäviä ja ohjelmoiden yksinkertaisia ohjelmia. Tehtäväesimerkki: *Ohjelmoi ohjelma, joka pyytää käyttäjältä kaksi lukua, kertoo ne ja ilmoittaa tulon.* Alla kuva esimerkikiratkaisusta.

Voit tutusta Python-ohjelmointikielen tehtäväpakettiin [täältä](#).

```
# Pyydetään käyttäjältä kaksi lukua.  
# Tallennetaan ne muuttujiin.  
Luku1 = input("Syötä luku: ")  
Luku2 = input("Syötä toinen luku: ")  
  
# Käsittellään lukuja kokonaislukuina.  
# Tallennetaan niiden tulo muuttujaan.  
Tulo = int(Luku1) * int(Luku2)  
  
# Ilmoitetaan tulo. Käsittellään muuttujaa merkkijonona.  
print("Lukujen tulo on " + str(Tulo))
```

```
Etunimi = "Samuel" # Merkkijono. Englanniksi string (str).  
Ika = 17           # Kokonaisluku. Eng. integer (int).  
Pituus = 1.74      # Liukuluku. Eng. float (float).  
Harrastukset = ["Golf", "Videopelit"] # Lista. Eng. list (list).  
Asuu_kotona = True # Totuusarvo. Eng. boolean (bool).
```

Esimerkkejä tietotyypeistä Python-koodina.



Tietotyypit ovat usein yksi etenemistä estävistä kompastuskivistä koodausta aloittavalle. On tärkeää ymmärtää, miksi numerot ja teksti eivät ole suoraan yhdistettävissä toisiinsa.



10. Itsenäistä harjoittelua verkossa

Opitaan itsenäisesti eri ohjelmointikielien perusteita. Toiminta sopii hyvin opettajan itseopiskeluun sekä oppilaiden itsenäiseen työskentelyyn opettajan avustuksella. Sivusto on englanninkielinen.

[W3schools.com](https://www.w3schools.com) tarjoaa maksutta opastuksia eri ohjelmointikielien perusteisiin. Klikkaa vain etusivulla yläpalkista "Tutorials" ja valitse mitä haluat oppia.

Vastaavia sivustoja on lukuisia, osa myös suomeksi. Pythonin perusteisiin voitte tutustua tie.koodariksi.fi -sivustolla, joka on Helsingin yliopistossa rakennettu maksuton oppimispolku Pythonin alkeiden itseopiskeluun.

[Koodaustunti](#) -sivustolle on koottu paljon tehtäviä, oppaita ja oppituntimalleja koodauksen opetukseen. Osa sisällöistä löytyy myös suomeksi.

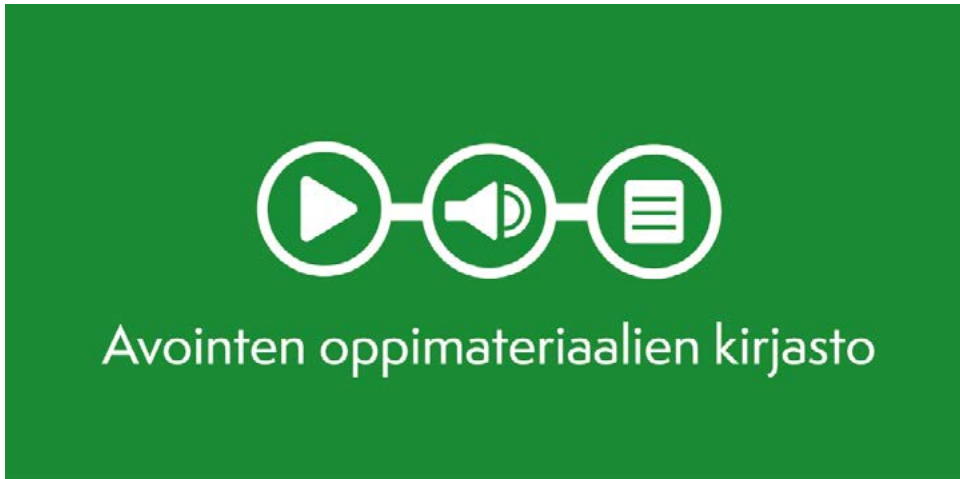
Englanninkieliselle [Khan Academy](https://www.khanacademy.com) -sivustolle on koottu erilaisia ohjelmointikursseja.





11. Avoimia oppimateriaaleja ohjelmointiin

Saatavilla on paljon tietoa, työkaluja ja oppaita ohjelmoinnin opettamiseen. Voit tutustua olemassa oleviin materiaaleihin erityisesti oman opetuksesi tavoitteiden sekä ryhmäsi näkökulmasta. Sisällöt löytyvät Avointen oppimateriaalien kirjastosta osoitteesta aoe.fi.



- ◆ ["Ohjelmointia Scratchin kanssa: Opas aloittelevalle ohjelmoijalle"](#) tarjoaa toimintaohjeet aloittelevalle Scratchin käyttäjälle. Oppaan avulla voidaan aloittaa perehtyminen myös vaativimpiin ohjelmointikieliin – niiden taustalla olevaan logiikkaan.
- ◆ ["Arduino-projekteja Maker kitillä"](#) -kurssilla rakennetaan ja ohjelmoidaan erilaisia virtapiirejä ja prototyyppisiä Arduino-alustaa käyttäen. Kurssi sopii monimuoto-opetukseen ja itseopiskeluun mutta harjoitusten tekeminen vaatii Arduino-alustan ja kokoelman peruskomponentteja. Kurssia on käytetty lukioiden lisäksi yläkouluissa.
- ◆ [Ohjelmointia matematiikan opetukseen](#) -kirjanen sisältää kokoelman tehtäviä, jotka yhdistävät ohjelmointia ja matematiikkaa eri tavoin. Kokoelman tarkoituksena on antaa ideoita peruskoulun ja lukion matematiikan tunneille ohjelmoinnin sovelluksista.
- ◆ Peruskoulun opettajille suunnatun [Let Me Hack It - Luovan teknologian ja ohjelmoinnin opettajamateriaalin](#) harjoitukset antavat ideoita luovuuden, teknologian ja pedagogiikan yhdistämiseen sekä monialaisten oppimiskokonaisuuksien suunnitteluun.
- ◆ Scratch ja Python oppimispaketit jakautuvat viiteen osaan: Lause, Silmukka, Ehtolause, Muuttuja ja Lista. Jokaisessa osassa on teoriaosuus ja haasteita, joilla voit kokeilla osaamistasi ja harjoitella ohjelmointitaitoja. Voit tehdä kaikki tehtävät joko pelkästään Scratch-ympäristössä, Python-ympäristössä tai sitten näitä kahta luovasti yhdistellen. Tehtävät ovat siis aivan tai lähes vastaavia.
 - » [Python from Scratch](#) -ohjelmoinnin oppimispaketti Python-ympäristössä. Materiaalin rinnalla kulkee koko ajan Scratch-materiaalit.
 - » [Scratch to Python](#) -kurssilla opitaan ohjelmointia Scratchilla. Materiaalin rinnalla kulkee koko ajan Python-materiaalit.

2

Tutkiva työskentely ja tuottaminen



Tutkiva työskentely ja tuottaminen on osa ohjelmointiosaamista. Osa-alueen osaamisen kuvaukset ja esimerkit keskittyvät ohjelmoinnin käyttämiseen työkaluna eri oppiaineissa sekä oppiainerajoja ylittävissä projekteissa. Tutkiva työskentely ja tuottaminen pohjautuu yhteiseen kehittelyyn, luovaan tuottamiseen sekä ohjelmointiin oppimisen välineenä. Oikean maailman ilmiöitä ja eri oppiaineiden sisältöä visualisoidaan, mallinnetaan ja dokumentoidaan ohjelmoinnin avulla. Painotetaan yhdessä tuottamista, prosessinomaista työskentelyä sekä palautteen antamista ja saamista. Osaamisen kuvausten mukaiseksi työskentelyksi sopii niin lautapelien tekeminen kuin esineitä tunnistavan algoritmin ohjelmointi.

Hyvä osaaminen vuosiluokilla 7–9

Yhteiskehittelyn prosessit

Oppilas ymmärtää erilaisia ryhmärooleja ja yhteistyön tapoja sekä työskentelee vastavuoroisesti ja aktiivisesti osallistuen ohjelmointiprojekteissa.

Oppilas osaa jakaa omia ohjelmoituja tuotoksiaan muiden käytettäväksi sekä osaa hyödyntää muiden tekemiä tuotoksia vastuullisesti.

Luova tuottaminen

Oppilas osaa suunnitella ja toteuttaa yhteistyössä muiden kanssa ratkaisun, jossa käytetään jotakin sensoreita hyödyntävää ratkaisua sekä robotiikkaa tai automatisaatiota.

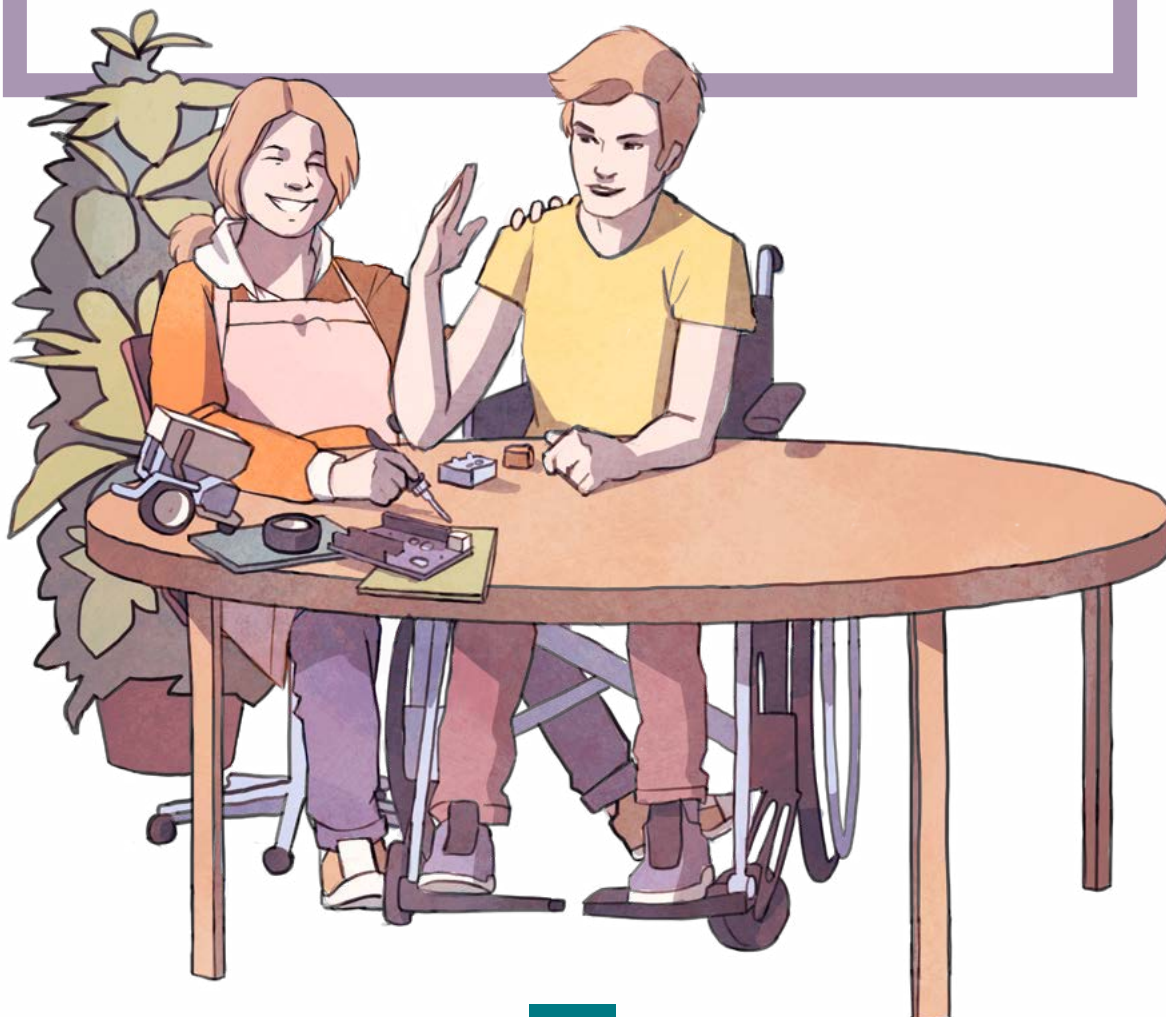
Oppilas osaa suunnitella ja toteuttaa pelin, simulaation tai sovelluksen, joka ratkaisee jonkin kouluun tai oppilaan omaan elämään liittyvän ongelman.

Ohjelmointi oppimisen välineenä

Oppilas tuntee eri oppiaineisiin liittyviä teknologisia sovelluksia ja osaa selittää niiden toimintaperiaatteita. Oppilas osaa hyödyntää ohjelmointiin liittyviä työskentelytapoja kaikissa oppiaineissa.

Käytännön taidot

Oppilas osaa ohjelmoida ohjelmia eri ympäristöissä sekä tuntee perusasiat yhdestä tekstipohjaisesta ohjelmointikielestä.





12. Tehdään lautapelejä

Opitaan suunnitelmallista työskentelyä yhteiskehittelyn hengessä, peliprojektin valmiiksi saamiseksi. Tuotetaan ja kehitetään omia luovia tuotoksia ideoiden, kokeillen ja palautetta hyödyntäen.

Peliprojektit noudattavat monin tavoin ohjelmoinnillisen ajattelun periaatteita. Perehtykää lautapeleihin ryhmäkeskustelun ja tiedonhaun kautta. Tehkää 3–4 hengen ryhmissä omia ajankohtaiseen teemaan tai oppiaineeseen sopivia lautapelejä. Suunnitelkaa pelin säännöt, tehkää pelilaudat ja muut materiaalit sekä testatkaa toistenne pelejä. Esitelkää projekteja eri vaiheissa prosessia antaen palautetta toisten projekteista.

Voitte käyttää lautapelin suunnittelussa vuokaavioita (katso **Yleistetään ilmiötä kaavoilla** -esimerkki). Esimerkiksi yksittäisen vuoron, kierroksen ja kokonaisen pelin kuvaaminen erillisillä kaavioilla auttaa sääntöjen suunnittelussa.

Vinkki!

Projekti voidaan toteuttaa monialaisena oppimiskokonaisuutena, jossa yhdistetään esimerkiksi äidinkieltä, käsityötä ja kuvataidetta. Myös robotiikkasarjoja ja kehitysalustoja voidaan hyödyntää esimerkiksi sääntöjen valvonnassa tai pisteiden laskijoina.



13. Herätetään maalaus eloon

Opitaan ohjelmoinnin luovaa käyttöä osana taideainetta. Yhdistellään fyysistä taidetta ja logiikkaa digitaalseksi teokseksi. Ideoidaan ja työskennellään yhteisten tavoitteiden saavuttamiseksi.

- ◆ Tässä esimerkissä hyödynnetään Scratch-ohjelmointiympäristöä. Löydät valmiin [malliprojektin täältä](#).

Tuottakaa parityöskentelynä taidetta välittämälläne työtavalla paperille. Toinen oppilaista tekee taustan yhdelle paperille ja toinen tekee liikkuvaksi animoitavan kuvan toiselle paperille.

Ottakaa kuvat teoksista ja tuokaa ne tietokoneelle. Pyyhikää liikkuvaksi tarkoitettua teoksesta kaikki ylimääräinen pois. Siirtäkää molemmat kuvat Scratchiin muokattavaksi, ohjelmoitavaksi ja animoitavaksi. Taustakuva asetetaan Scratchiin taustaksi. Animoitavaksi tarkoitettu kuva asetetaan Scratchissa hahmoksi ja sen asusteista tehdään kopioita. Muokatkaa kopiota, jotta saatte näkyviin hahmon muutoksen tai liikkeen.





14. Tehdään yhdessä pelejä

Opitaan suunnitelmallista työskentelyä yhteiskehittelyn hengessä ohjelmointiprojektin valmiiksi saamiseksi. Tuotetaan ja kehitetään omia luovia tuotoksia ideoiden, kokeillen ja palautetta hyödyntäen.

Hakekaa tietoa peleille ominaisista piirteistä ja pelimekaniikoista. Keskustelkaa tuloksista yhdessä ja kerätkää ne yhteen. Tehkää löydettyjä ominaisuuksia hyödyntäen 2–4 hengen ryhmissä pelejä – oppimispelejä tai puhtaasti viihdepelejä. Pelin tekemisen sijaan voitte myös suunnitella pelin konsepteja opetuksen ajankohtaisten sisältöjen ja tavoitteiden mukaisesti.

Kukin ryhmä tekee suunnitelman pelistään ennen toteuttamista. Jakakaa roolit, joita voidaan välillä vaihtaa ryhmän sisällä:

- ◆ grafiikan piirtäminen,
- ◆ musiikki ja ääniefektit,
- ◆ pelimekaniikan yksityiskohtien miettiminen,
- ◆ toteutus koodiksi.

Esitelkää projekteja eri vaiheissa prosessia antaen palautetta toisten projekteista ja myös kehittäen työtä niiden pohjalta.

Voitte käyttää pelin suunnittelussa vuokaavioita (katso **Suunnitellaan algoritmeja** -esimerkki). Esimerkiksi yksittäisen vuoron, kierroksen ja kokonaisen pelin kuvaaminen erillisillä kaavioilla auttaa sääntöjen suunnittelussa.

Löydät esimerkin Scratchilla tehdystä [kertolaskupelistä täältä](#).

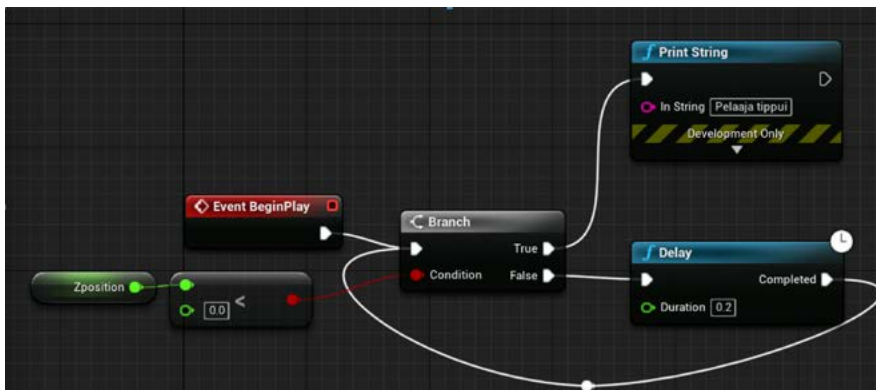
Vinkki!

Pelien tekemiseen soveltuu esimerkiksi [Scratch](#), [GDevelop5](#) ja [Pythonin Pygame-kirjasto](#). Voitte käyttää pelien tekoon myös 3D-ympäristöjä. Ne ovat motivoivia ja ammatillaisia välineitä, mutta vaativat opettajalta paljon perehtymistä.



Pelintekoympäristöjä on erilaisia. Esimerkiksi Unity, Unreal Engine ja Godot ovat 3D-pelimootoreita, jotka sopivat sekä harrastus- että ammatikäyttöön. GDevelop on 2D-pelien tekemiseen suunnattu pelintekoympäristö.

- » [Unity](#)
- » [Unity-oppimisalusta](#)
- » [Unreal Engine](#)
- » [Unreal Engine -oppimisalusta](#)
- » [Godot](#)
- » [GDevelop 5](#)



Useat 3D-pelinteko-ohjelmat tarjoavat vuokaavioiden tyyllisiä vaihtoehtoja pelien ohjelmointiin. Kuvassa esimerkki Unreal Engine:n Blueprint-ohjelmoinnista.



15. Kilpaillaan roboteilla

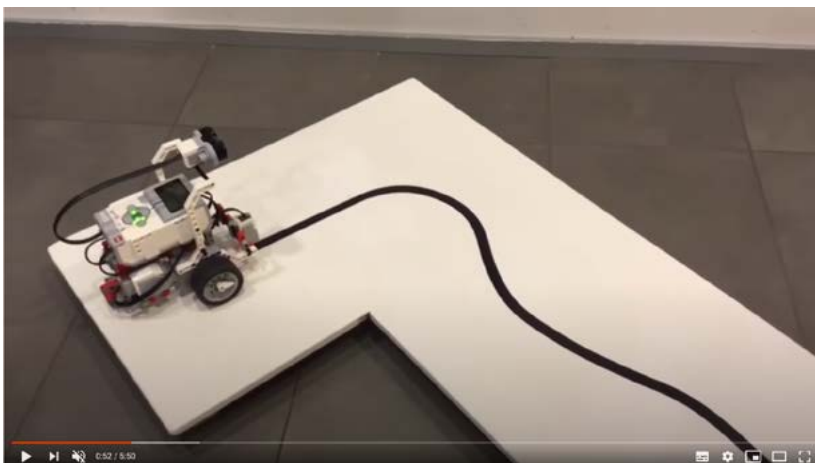
Opitaan käyttämään ohjelmoinnille ja innovoinnille tyypillistä iteratiivista kehitysprosessia sekä tuottamaan toimivaa automaatiota. Kehitetään omia automaattisia laitteita kehitysalustaa hyödyntäen. Ideoidaan ja työskennellään yhteisten tavoitteiden saavuttamiseksi. Kehitetään yhdessä kilpailu, jossa on säännöt.

- ◆ Esimerkki pohjautuu robotiikkasarjan käyttämiseen.

Tutustukaa käytössä olevan robotiikkasarjan komponentteihin, ohjelmointiin ja mekaniikkaan. Ideoikaa yhdessä erilaisia kilpailuja, joissa robotit voisivat mittelöidä. Voitte ottaa inspiraatiota oikeista urheilulajeista. Äänestäkää 1–3 lajia, joita varten oppilaat suunnittelevat ja toteuttavat robottinsa. Suunnitelkaa ja rakentakaa 2–3 hengen ryhmissä mahdollisimman kilpailukykyiset robotit. Projektin toteutukseen sopivat esimerkiksi LEGO EV3-, VEX IQ- sekä muut vastaavat robotiikkasarjat. Voitte aluksi käyttää robotiikkasarjojen mukana tulevia rakennusohjeita ja jalostaa sen pohjalta robotit ryhmien tehtäviin sopiviksi ja ryhmän näköisiksi.

Vinkki!

Yksi suosittu robottien kilpailu on Sumokilpailu, jossa vastakkain olevat robotit pyrkivät työntämään toisen robotin ulos kilpailualustalta. Voit tutustua sumorobottien toimintaan videolla.



Sumorobotit:

https://youtu.be/G_v8SR9Xp04

Lähde: Innokas-verkosto. Lisää tietoa Innokas-verkoston toiminnasta löydät osoitteesta www.innokas.fi.

Vinkki!

Toinen toimivaksi todettu laji on "robotti-imurit": Levittäkää rajatulle alueelle metallisia aluslevyjä tai muita kevyitä metalliesineitä, jotka tarttuvat magneetteihin. Kiinnittäkää robottiin pieniä magneetteja. Oppilaiden tehtävä on tehdä robotille ohjelma, joka kerää mahdollisimman paljon kentällä olevia aluslevyjä 30 sekunnissa.



16. Automatisoidaan Minecraftissa

Opitaan automatisointia ja ohjelmoinnin logiikkaa peliympäristössä. Kehitetään omia automaattisia laitteita digitaalisen ympäristön lainalaisuuksien vallitessa. Ideoidaan ja työskennellään yhteisten tavoitteiden saavuttamiseksi. Toiminnassa korostuu oppilaiden oma aikaisempi osaaminen, tiedonhaku ja ideoiden jakaminen.

- ◆ Toimintaa varten tarvitsette [Minecraft Education Editionin](#) tai tavallisen [Minecraftin](#)

Tehkää ja rakentakaa pienissä 2–3 hengen ryhmissä Minecraftissa erilaisia ansoja ja koneita punakivellä (redstone). Pelimuodoksi kannattaa valita nikkaroija (perustasoissa Minecraftissa nimellä ”luova”). Punakivijauheen, -soihtujen ja -toistimien avulla rakentamisen mahdollisuudet ovat lähes rajattomat. Hyviä pieniä projekteja ovat esimerkiksi ansat sekä maanviljelyn ja tavaran varastoinnin automatisointi. Esitelmää ja testatkaa toisten tuotoksia. Laitteiden algoritmien suunnittelussa voitte hyödyntää vuokaavioita (katso **Suunnitellaan algoritmeja** -esimerkki).

Vinkki!

Projekti voidaan skaalata ylöspäin myös pakuhuone-projektiksi, jolloin koko luokka rakentaa sarjan pulmatehtäviä suljettuun tilaan samalla palvelimella.

Konkreettisia esimerkkejä:

- ◆ Tee painelaatasta aktivoituva laavakuoppa-ansa, jonka voi kytkeä päälle ja pois päältä vipua kääntämällä (pinalaatta, punakiveä, punakivisohtuja, punakivitoistimia, mäntiä).
- ◆ Tee ansahuone, jonka lattia on painelaattojen peitossa. Osa painelaatoista aktivoi ansan (esim. katon laskeutuminen), mutta huoneen läpi on yksi turvallinen reitti.



Pelillinen haaste ja luova tekeminen yhdistyvät ohjelmointilogiikan harjoitteluun, sillä Minecraftin punakivisignaalit vastaavat lähes tulkoon fyysisiä elektronikkassa käytettyjä johtimia ja puolijohteita. Jos oppilaat pystyvät rakentamaan and- ja or-portteja, voivat he rakentaa mitä tahansa loogisia kytkentöjä. Nykyiset tietokoneet sisältävät miljardeja vastaavia logiikkaportteja, mutta toimivat samalla perusperiaatteella.





17. Ohjelmoidaan kielentunnistusalgoritmi

Opitaan tekstipohjaisen ohjelmointikielen käyttämistä sekä siihen liittyvää koodin korjaamista ja ongelmanratkaisua. Ratkaistaan ongelmia aktiivisesti omien havaintojen perusteella. Opitaan peräkkäisyyden, toistuvuuden ja ehdollisten toimintojen käyttämistä.

Työskennelkää pareittain ja valitkaa kotimaisen kielen lisäksi 2–3 vierasta kieltä ja etsikää verkosta tekstejä kyseisillä kielillä. Kerätkää kyseisellä kielellä kirjoitettuja tekstikappaleita tekstidokumenttiin. Vertailkaa tekstejä ja pyrkikää löytämään teksteistä kielelle ominaisia piirteitä, jotka auttavat tunnistamaan kielen, vaikkei sitä osaisikaan tulkita. Esitelkää havainnot toisille oppilaille. Kielelle ominaisia piirteitä ovat esimerkiksi erikoiskirjaimet, yksittäiset yleiset sanat ja päätteet.

Tehkää tekstipohjaisella ohjelmointikielellä ohjelma, joka erottaa tutkitut kielet toisistaan sille syötettyä tekstikappaletta analysoimalla. Ohjelma siis yleistäen arvaa, mistä kielestä on kyse kielille tyyppisiä merkkejä ja sanoja etsimällä. Oman algoritmin suunnittelussa kannattaa hyödyntää vuokavioita (katso **Suunnitellaan algoritmeja** -esimerkki).

Vinkki!

Voitte käyttää esimerkkinä tai pohjana valmista [Python-ohjelman pohjaa](#), joka tunnistaa tekstin perusteella suomen, ruotsin ja englannin kielen.



Maailman säännönmukaisuuksien tutkiminen on jo itsessään olennaista, mutta se on sitä myös tietokoneiden ymmärtämisen näkökulmasta. Nykyään eri algoritmit osaavat jo erinomaisesti tunnistaa eri kieliä erityisesti tekstin muodossa, mutta myös yhä paremmin puhuttuna. Omaa ohjelmaa tehdessä huomataan, että ohjelmalle syötetyn tekstin määrä lisää mahdollisuutta onnistua kielen tunnistuksessa.



18. Tehdään interaktiivisia esitelmiä ilmiöistä

Opitaan suunnitelmallista työskentelyä yhteiskehittelyn hengessä ohjelmointiprojektin valmiiksi saamiseksi. Tuotetaan ja kehitetään omia tuotoksia ideoiden, kokeillen ja palautetta hyödyntäen.

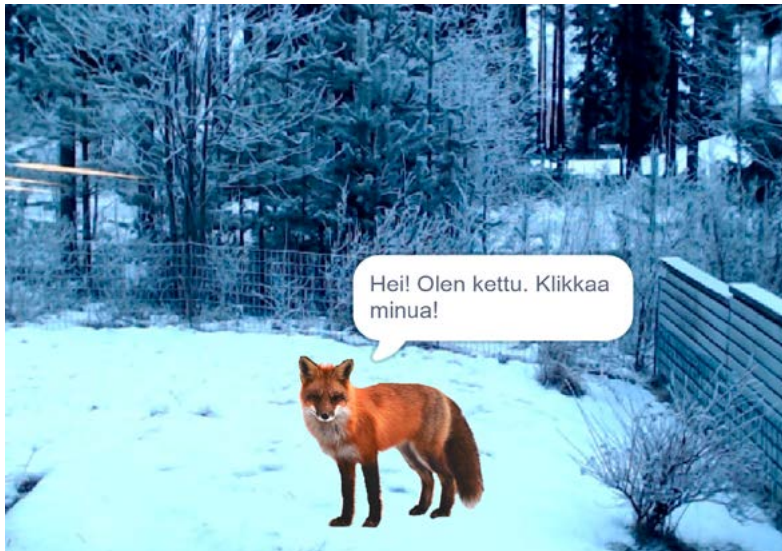
Tehkää pareittain tai pienissä ryhmissä ajankohtaisiin oppiaineen ilmiöihin tai aiheisiin liittyviä interaktiivisia mallinnuksia, visuaalisia kuvauksia tai esitelmiä. Esimerkiksi kemiassa voitte tehdä Bohrin atomimallin, jossa elektronikuoret pyörivät ytimen ympärillä. Biologiassa taas voidaan tehdä visuaalinen kuvaus esimerkiksi veden kiertokulusta. Tarkoitus on tehdä ilmiötä kuvaava malli, joka sisältää pientä interaktiivisuutta. Käyttäjä voi siis vaikuttaa ohjelman toimintaan esimerkiksi klikkaamalla painikkeita. Esitelkää projekteja eri vaiheissa prosessia antaen palautetta toisten projekteista.

Vinkki!

Kaksi seuraavaa esimerkkiä on toteutettu Scratch-ympäristössä.

[Esimerkki 1: Atomimalli](#)

[Esimerkki 2: Kettu](#)





19. Ohjelmoidaan esineitä tunnistava algoritmi

Opitaan käyttämään ohjelmoinnille ja innovoinnille tyypillistä iteratiivista, eli toistuvaa kehitysprosessia sekä tekemään toimivia ohjelmia graafisessa ohjelmointiympäristössä. Jaetaan ideoita ja ratkaistaan yhdessä ongelmia yhteisten tavoitteiden saavuttamiseksi. Opitaan peräkkäisyyden, toistuvuuden ja ehdollisten toimintojen käyttämistä. Ohjelmointiympäristössä voidaan käsitellä konenäön käyttöä ja toimintaperiaatteita.

Toimintaa varten tarvitsette Scratch-ohjelmointiympäristön ja [videohavainnointi-laajennuksen](#).

Ohjelmoikaa pareittain tai pienissä ryhmissä Scratchin Videohavainnointi-laajennusta hyödyntäen algoritmi, joka tunnistaa 3–5 esinettä. Ohjelman toiminta perustuu ehtolauseisiin, joissa on esineelle ominaisia värejä yhdistettynä tai-operaattoriin. Jos esineeseen liittyvä väri löytyy, kasvatetaan kyseisen esineen "osumat"-muuttujaa. Tunnistetun esineen "osumat"-muuttuja on arvoltaan suurin.

Projekti on haastava, mutta tarjoaa mahdollisuuden ohjelmoida konenäköä itse. Mitä enemmän tunnistettavia esineitä on, sitä monimutkaisemmaksi ohjelma muuttuu. Oppilas tai edes opettaja ei välttämättä yksin keksi, miten ohjelman saa tunnistamaan useita esineitä. Tähän tarvitaan ryhmän yhteistä panosta ja ideoiden jakamista.

Vinkki!

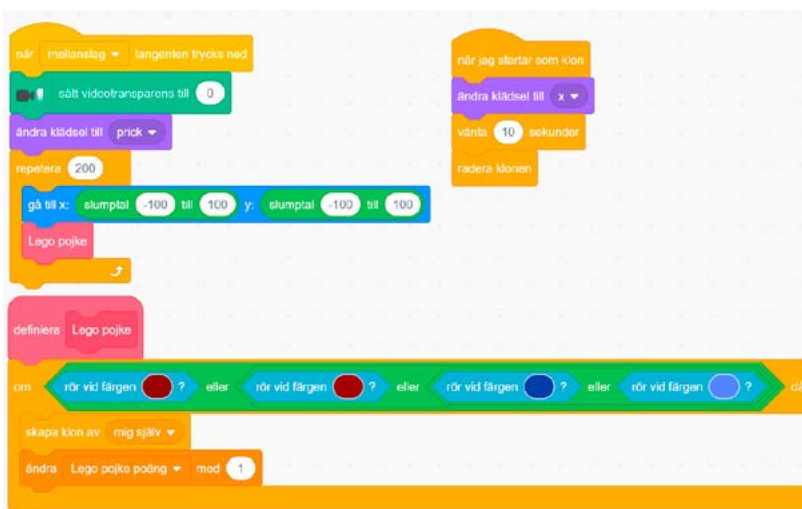
Voitte käyttää pohjana valmista [esineentunnistusohjelmaa](#).

Vinkki!

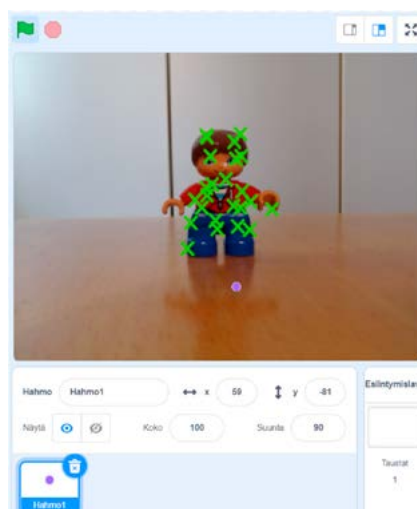
Projektin voi yhdistää esimerkiksi ympäristöopin kasvintunnistukseen (kasvit, jotka ovat selkeästi eri värisiä) tai käsitöissä robotiikkaan lisäämällä myös esimerkiksi LEGO EV3 -laajennuksen.



Nykyteknologiassa konenäköä hyödynnetään enenevissä määrin. Yksi tutuimmista esimerkeistä on älypuhelimien kasvojentunnistus.



Kuvassa esimerkki koodista, jolla löydetään yhden esineen värejä kamerakuvasta. Tunnistettavia esineitä voidaan lisätä tekemällä uusia funktioita (omia lohkoja) kuten yllä olevassa kuvassa näkyvä "LEGO-poika" -lohko. Uudelle esineelle täytyy myös tehdä oma "osumat"-muuttuja.



Ohjelmoitava hahmo on pieni piste, jonka toinen asuste on rasti. Aina kun esineeseen liittyvä väri löydetään, jätetään sille hahmon kloonin, joka muuttuu rastiksi ja häviää 10 sekunnin kuluttua.



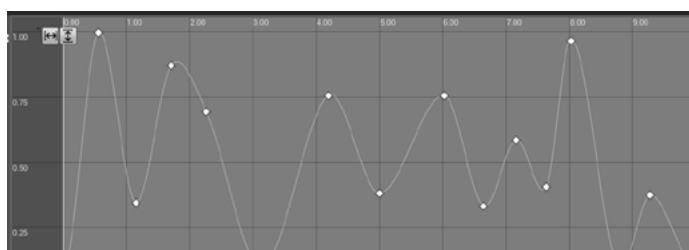
Tämä harjoitus on haastava 3D-ympäristöjen monimutkaisuuden vuoksi, mutta toisaalta 3D-ympäristöt tarjoavat valtavasti potentiaalia ja motivoivaa tekemistä.



- » Unity
- » Unity-oppimisalusta
- » Unreal Engine
- » Unreal Engine
-oppimisalusta
- » Godot



Jos halutaan luoda luonnollinen valonlähte, voidaan ottaa referenssiksi esimerkiksi oikea kynttilä. Valon värin säätäminen on helppoa, mutta entä luonnollinen välkehtiminen ja epätasainen palaminen?



Valon kirkkaus noudattaa graafia, jossa on satunnaiselta vaikuttavaa vaihtelua.



21. Luodaan arjen automaatiota

Opitaan käyttämään ohjelmoinnille ja innovoinnille tyypillistä iteratiivista kehitysprosessia sekä tekemään toimivia ohjelmia graafisessa ohjelmointiympäristössä. Opitaan automaation peruseräiteista. Opitaan havainnoimaan ja tekemään tulkintoja arjen automaatiosta. Ideoidaan, annetaan palautetta ja työskennellään yhteisten tavoitteiden saavuttamiseksi. Kehitetään yhdessä ratkaisuja arjen ongelmiin hyödyntäen kehitysalustoja.

1. Etsikää lähiympäristöstä sähkölaitteita ja ottakaa niistä kuvia. Luokitelkaa kuvat automaattisiin ja manuaalisiin laitteisiin (esimerkiksi automaattisesti vai katkaisimesta syttyvät valot). Pohtikaa yhdessä, mitä apua automaattisuudesta on. Tehkää kuvakollaasi automaattisista laitteista.
 - » Esimerkiksi automaattiset ovet, valot ja käsipyyhkeen annostelijat löytyvät jo melkein kaikilta kouluilta, mutta niitä voidaan löytää myös esimerkiksi kirjastosta tai muista julkisista tiloista.
2. Voitte syventää toimintaa eteenpäin robotiikalla. Tuottakaa tai suunnitelkaa pareittain tai pienissä ryhmissä kouluun laitteita, joilla ratkaistaan arjen puutteita tai ongelmia. Laitteiden algoritmien suunnittelussa kannattaa hyödyntää vuokaavioita (katso **Suunnitellaan algoritmeja** -esimerkki).
 - » Voitte suunnitella esimerkiksi robotiikkaa ja eri materiaaleja yhdistämällä käsidesiannostelijoita, juoma-automaatteja ja valokatkaisimia käyttäviä robotteja. Kehitysalustoiksi sopivat robotiikkasarjat (esimerkiksi LEGO EV3 ja VEX IQ) sekä mikrokontrollerit (esimerkiksi Micro:bit ja Arduino).

Vinkki!

Laitteiden kehittämisen eri vaiheissa oppilaille on hyvä antaa mahdollisuus esitellä ideansa ja tuoksensa. Näin mahdollistetaan palautteenanto ja yhteiskehittäjyden prosessin näkyväksi tuominen.





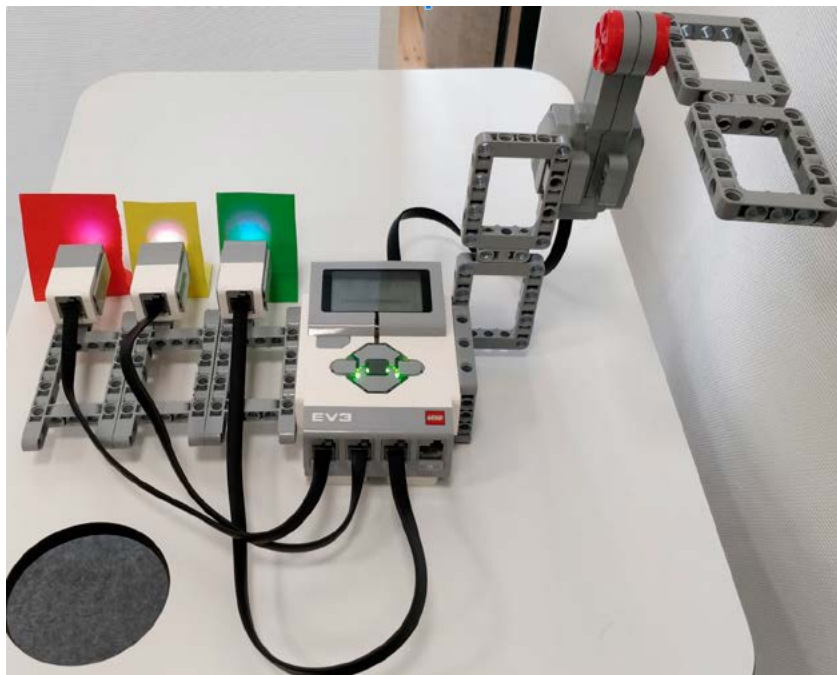
22. Rakennetaan pakohuone

Opitaan käyttämään ohjelmoinnille ja innovoinnille tyypillistä iteratiivista kehitysprosessia sekä tuottamaan toimivaa automaatiota. Kehitetään omia automaattisia laitteita kehitysalustaa hyödyntäen. Ideoidaan ja työskennellään yhteisten tavoitteiden saavuttamiseksi. Kehitetään yhdessä automatisoitu projektikokonaisuus.

Tehkää yhdessä [pakohuonekokonaisuus](#), jossa oppilaiden tehtävänä on pienissä ryhmissä luoda pulmatehtäviä. Kytkekää kokonaisteema opiskeltavaan ilmiöön tai opetussisältöön. Jokainen pulmatehtävä antaa ratkaistuna palan salasanaa, jolla pääsee pois pakohuoneesta. Jokainen pulma sisältää arvoituksen, joka ratkaistaan jollain syötteellä, esimerkiksi painamalla painikkeita oikeassa järjestyksessä tai näyttämällä värianturille oikean väristä esinettä tai vastaamalla oikein johonkin arvoitukseen. Voitte myös rakentaa fyysisiä lavasteita, vihjeitä ja esteitä. Laitteiden algoritmien suunnittelussa kannattaa hyödyntää vuokaavioita (katso **Suunnitellaan algoritmeja** -esimerkki).

Vinkki!

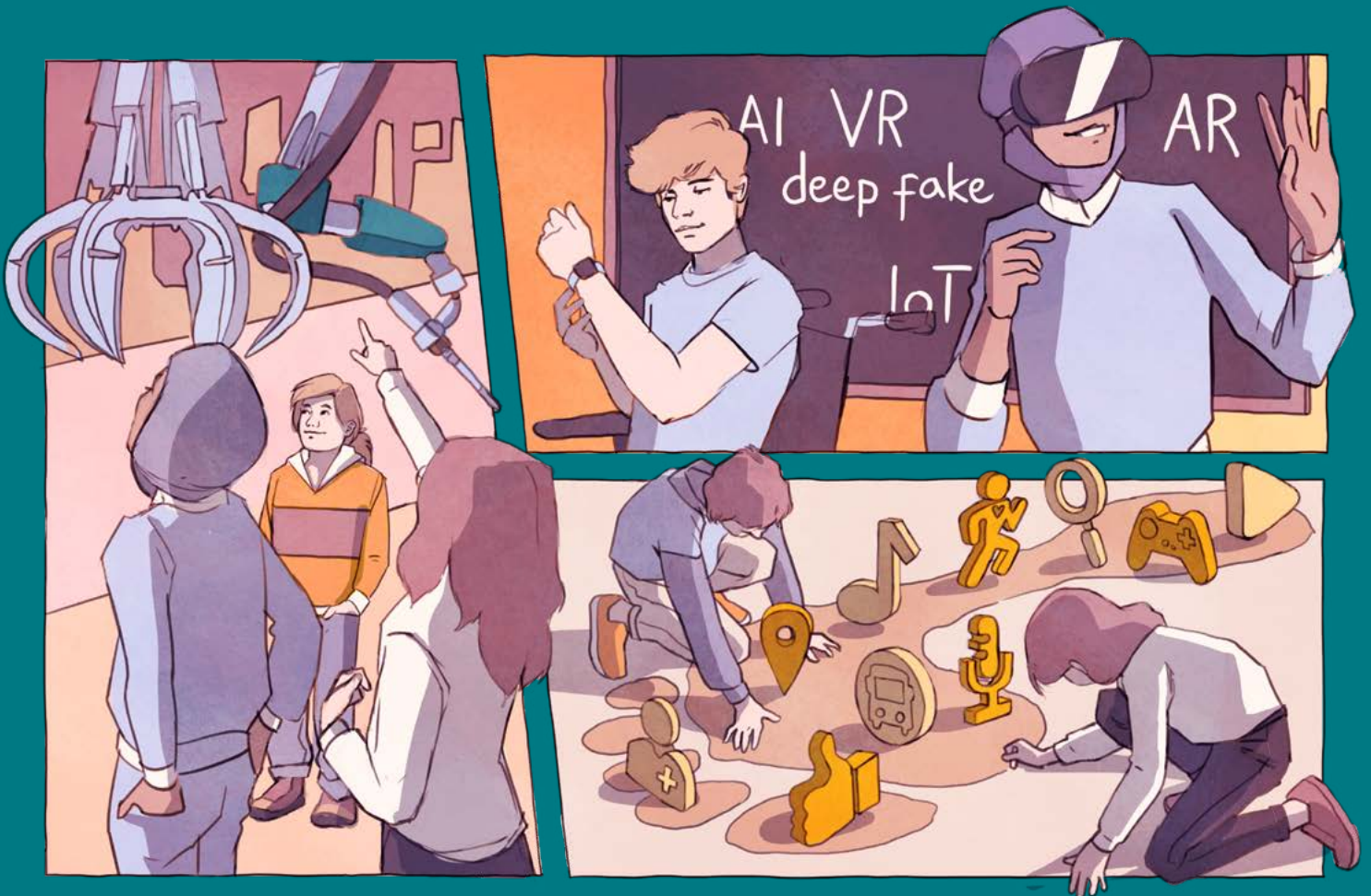
Projekti voidaan toteuttaa monialaisena oppimiskokonaisuutena, jossa yhdistetään esimerkiksi käsityötä, kuvataidetta ja lukuaineiden sisältöjä.



Esimerkki yksittäisestä pulmasta. Kolmelle värianturille täytyy syöttää oikeaa väriä, jolloin pulma on ratkaistu ja pelaajat saavat palan salasanaa (luukku aukeaa tai teksti ilmestyy keskusyksikön näytölle). Yksi pulma voisi liittyä esimerkiksi kemiaan: Pöydällä on 10 eriväristä korttia, joissa on alkuaineiden lyhenteitä. Pulman vihjeenä olisi "rakenna natriumhydroksidi" ja pelaajien pitäisi asettaa järjestykseen kortit, joissa lukee "Na", "O" ja "H".

3

Ohjelmoidut ympäristöt ja niissä toimiminen



Digitaalisuus koskee yhä vahvemmin ihmisten arjen, yhteiskunnan ja kulttuurin osa-alueita. Ohjelmoinnilla ja koodilla voidaan luoda erilaisia ympäristöjä, edellytyksiä ja mahdollisuuksia, mutta samalla ne voivat myös luoda rajoitteita ja osaltaan haasteita ihmisten elämään. Yläkoulun ohjelmointiosaamisessa pyritään ottamaan vahvemmin esille ohjelmoitujen ympäristöjen eettistä ja yhteiskunnallista merkitystä. Algoritmit ja niihin pohjautuva profilointi vaikuttavat suunnitellusti tai tahattomasti ihmisten toimintaan esimerkiksi valikoimalla tietoa, luomalla yhteyksiä ja suuntaamalla toimintamahdollisuuksia. Digitaalinen maailma korostaa ymmärrystä henkilötiedoista ja muusta tiedosta. Julkishallinto, yritykset ja erilaiset järjestöt sekä yhteisöt keräävät ihmisistä paljon erilaista tietoa. Ihmiset myös luovuttavat ja julkaisevat itsestään paljon asioita ja tietoa esimerkiksi sosiaalisen median palveluita käyttäessään.

Ymmärrys tietotekniikasta, internetistä, tiedosta ja ohjelmoiduista ympäristöistä sekä niiden toiminnasta auttaa myös kehittämään tietoturvaa sekä huolehtimaan henkilökohtaisista tiedoista. Ohjelmointiosaamisessa ja digitaalisten rakenteiden ymmärtämisessä on kyse myös tasa-arvosta, demokratiasta ja valtasuhteista: Taitojen ja ymmärryksen epätasapaino voi johtaa osallisuuden eriytymiseen.



Hyvä osaaminen vuosiluokilla 7–9

Ohjelmoitu teknologia elämän eri osa-alueilla

Oppilas tuntee ohjelmoituja elementtejä ympäröivässä yhteiskunnassa, kuten algoritmi, automaatio, robotiikka ja tekoäly. Oppilas ymmärtää niiden toimintalogiikkaa ja sovelluksia elämän eri osa-alueilla.

Ohjelmoidun teknologian vaikutukset arjessa

Oppilas ymmärtää, että kerätyt tiedot tallentuvat ja osaa antaa ainakin yhden esimerkin, mihin tarkoitukseen hänestä kerättyä tietoa voidaan käyttää.

Oppilas pohtii ohjelmoidun teknologian terveydellisiä, sosiaalisia, poliittisia, tieteellisiä ja käytännöllisiä mahdollisuuksia sekä riskejä ja eettisiä näkökulmia.



23. Haetaan tietoa ja opitaan hakemaan

Opitaan käyttämään hakukoneita tehokkaammin ja ymmärtämään niiden toimintaperiaatetta. Jaetaan kokemuksia ja ajatuksia teknologian käytöstä, myös eettisestä näkökulmasta.

Kilpailkaa, kuka löytää ensimmäisenä jonkin opiskeltavaan aihealueeseen liittyvän tiedonmurusen netistä. Keskustelkaa oikean vastauksen löydyttyä tiedonhakustrategioista. Mistä tieto löytyi, mitä hakukonetta käytettiin, millaisia hakusanoja käytettiin? Peilatkaa myös kokemuksianne ja pyrkikää jakamaan osaamista yhteisesti. Voitte koota vinkkejä ja toimivia toimintatapoja yhteen. Oliko tiedon etsiminen helppoa? Oliko löydettyissä tiedoissa tulkinnanvaraa? Millä keinoin tiedon luotettavuutta voi arvioida? Voitte jatkaa tehtävää vertailemalla eri hakukoneiden tuottamia tuloksia tai anonyymissa selaimessa saatuja tuloksia.

Esimerkkitehtäviä alla:

- » Selkeä vastaus: Missä on mitattu kaikkien aikojen lämpöennätys?
- » Selkeä vastaus: Mistä osista pöytätietokone tyypillisesti rakentuu?
- » Kysymyksen kanssa on oltava tarkkana: Montako presidenttiä USA:ssa on ollut? (eri presidenttikausien ja presidenttinä toimivien henkilöiden määrä ei ole sama)
- » Eri uskomukset vaikuttavat vastaukseen: Kuinka vanha maapallo on? Mitä eri käsityksiä tästä on?

Tehkää keskustelujen ja pohdintojen pohjalta hyvistä käytännöistä tietoiskuja tai koosteita muille oppilaille. Voitte tehdä pienissä ryhmissä vapaamuotoisia tuotoksia, joilla esittelette hakukoneiden luotettavaa käyttämistä. Tuotos voi olla esimerkiksi ohjejulist, mainos tai esitelmä, joka ripustetaan esille koulun käytävälle.



24. Kaikelle on syy

Opitaan palveluiden kriittistä tarkastelua tarkoituksiperän ja eettisyyden näkökulmasta.

Tehkää 2–3 hengen ryhmissä listoja tärkeimmistä palveluista (sovelluksista), joita oppilaat käyttävät sekä vapaa-ajalla että koulussa. Keskustelkaa sovelluksista yhteisesti ja pyrkikää löytämään niistä yhteisiä ja erottelevia piirteitä esimerkiksi niiden tarkoitusten mukaan. Miten luokittelisitte erilaiset sovellukset?

Palveluiden tuottaminen vaatii aikaa, osaamista, työtä ja erilaisia resursseja. Pohtikaa, millaisia motiiveja sovellusten takana on. Miten voi erottaa kaupallisen palvelun yleishyödyllisestä? Millaisia eri ansaintamalleja sovelluksilla voi olla? Millaisia eettisiä eroja sovellusten tarkoituksilla tai ansaintalogiikoilla voi olla?



Maksuttomien palveluiden kohdalla rahoitusmalleja voivat olla esimerkiksi seuraavat: a) pieni firma yrittää saada markkina-asemaa isoilta firmoilta, b) "freemium"-malli eli lisäominaisuuksista maksetaan, c) iso firma varmistaa asemaansa tarjoamalla maksuttomasti palveluita esim. opiskelijoille, d) mainostulot.

Kaikissa edellä mainituista tapauksista käyttäjän omat tiedot saatetaan kerätä ja niitä voidaan käyttää mainostamiseen, palvelun optimointiin tai myydä eteenpäin.

Ohjaa ryhmiä valitsemaan yksi ohjelma (esimerkiksi sovellus tai peli) ja tarkastelkaa ohjelman taustoja. Esitelmää tulokset yhteisesti. Pohtikaa esimerkiksi:

- » Kuka ohjelman on tehnyt?
- » Mikä ohjelman tarkoitus on?
- » Kenelle ohjelma on tarkoitettu? Mikä sen kohderyhmä on?
- » Millainen ansaintamalli ohjelmassa on?
- » Kuinka eettinen ohjelma on?





25. Kone vastaan ihminen

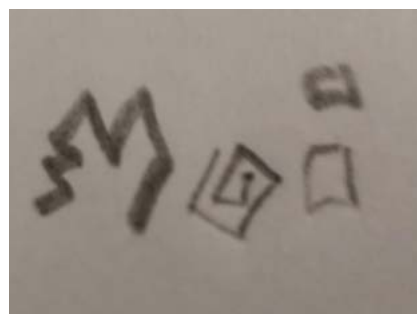
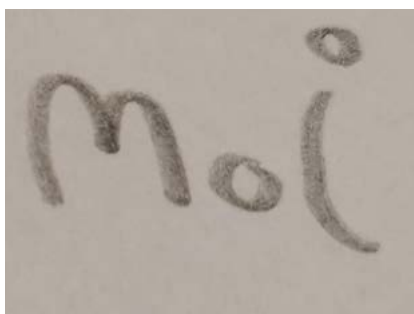
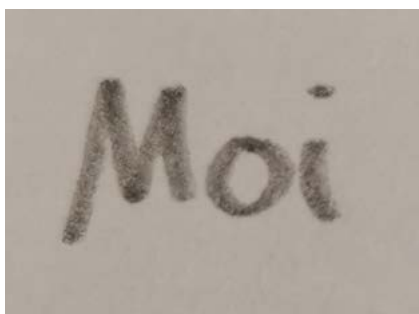
Opitaan tarkastelemaan teknologista maailmaa ja teknologian toimintaperiaatteita mahdollisuuksien ja rajallisuuden näkökulmasta.

- ◆ Toimintaa varten tarvitsette tekstiä kameran avulla tunnistavan sovelluksen tai ohjelman, kuten [Google lens](#) -sovelluksen.

Luokaa pareittain tai pienissä ryhmissä eri tavoin hankalasti kirjoitettua kirjoitusta (CAPTCHA-tekstejä), jota ihminen pystyy lukemaan, mutta jota tietokone ei ymmärrä. Kirjoittakaa tekstiä selkeästä vaikeampaan ja kokeilkaa välillä ohjelman tekstin tunnistusominaisuutta. Pohtikaa, miksi koneiden on laskentatehosta huolimatta haastavaa ymmärtää ihmisen luoman maailman ilmiöitä. Esitelmää havaintonne ja keskustelkaa yhdessä toiminnan mahdollisuuksista ja rajoitteista.



CAPTCHA = completely automated public turing test to tell computers and humans apart; käytännössä siis eri tavoin sotkettua tekstiä, josta tietokoneen on vaikea tunnistaa kirjaimia.



Tekoäly pystyy päivä päivältä parempiin tuloksiin. Kuvissa näkyvää kolmatta "Moi"-tekstiä ei tekoäly pystynyt tunnistamaan. Jos tekstiä jatkaa esimerkiksi "Moi, mitä kuuluu?", niin tunnistamisen todennäköisyys saattaa kasvaa, mikäli osa sanoista on selvästi tunnistettavissa. Myös selkeä teksti, jonka päälle on piirretty epätasaista viivaa, on helposti ihmisen tunnistettavissa, mutta tekoäly ei sitä välttämättä tunnista edes tekstiksi.



26. Sulautetut järjestelmät

Opitaan luokittelemaan ja havainnoimaan teknologiaa eri perustein. Opitaan ymmärtämään perusasioita laitteiden toimintaperiaatteista ja tarkoituksesta. Tiedetään, mitä sulautetut järjestelmät ovat. Havaitaan, että tietokoneita on yllättävän monessa paikassa piilossa.

Selvittääkää, mitä tarkoittaa sulautetun järjestelmän (**embedded system**) käsite. Kerätkää esimerkkejä sulautetuista järjestelmistä koulussa, kotona ja koulumatkalla. Mitkä laitteet ovat sulautettuja, mitkä taas eivät ole? Onko kaikissa nettiyhteys?

Tehkää ryhmissä eri aiheista kuvaa ja tekstiä yhdistäviä tuotoksia, jotka vastaavat esimerkiksi seuraaviin kysymyksiin (yksi kysymyspari = yksi tuotos):

- » Mistä sähkö tulee? Kuinka nopeasti sähkö liikkuu?
- » Miten internet toimii?
- » Mitä tietokoneeseen kuuluu? Miten sähkö liittyy tietokoneisiin?
- » Missä tietokoneita on? Mihin laitteet tarvitsevat antureita?
- » Mitä ovat IoT-laitteet? Mitä jos leivänpaahittimella olisi nettiyhteys?



Sulautettuun laitteeseen kuuluu antureita, tietokoneen prosessori tekemässä päätöksiä, sekä jonkinlainen vaikutus (näyttö, sähkömoottori, jonkin toiminnon käynnistyminen tai signaali toiselle laitteelle). Jos laitteessa on (langaton) nettiyhteys, on kyseessä **IoT-laitte** (internet of things).

Vinkki!

Voit auttaa oppilaita alkuun tiedonhaussa esimerkiksi tarjoamalla valmiita linkkejä, tulosteita, kirjoja ja apua englanninkielisten lähteiden kääntämisessä.

Vinkki!

Jos päädytte pohtimaan, voiko älykäs laite toimia ilman sähköä, voitte hakea Wikipediasta "[Analyttisen koneen](#)" artikkelin ja kuvan. Kyseessä on maailman ensimmäinen tietokone. Se ei tarvitse sähköä vaan on kokonaan mekaaninen. Mekaanisessa laitteessa tietyn signaalin kuljetus vaatii mekaanisia rakenteita. Tietokoneet toimivat sähköllä, koska signaalin lähetys sähkövirralla on nopeampaa, vaatii vähemmän liikkuvia osia ja kestää paremmin.



27. Robottiikan kolme sääntöä

Opitaan tarkastelemaan automaatiota ja robotiikkaa eettisyyden näkökulmasta ja jatkeena ihmisen omille arvoille. Tehdään teknologiaan liittyviä yleistyksiä sekä vertaillaan omia ajatuksia ja havaintoja toisten ajatuksiin ja havaintoihin.

Hakekaa tietoa Isaac Asimovin robotiikan kolmesta säännöstä. Kootkaa säännöt yhteen sekä pohtikaa ja keskustelkaa niistä yhdessä. Riittävätkö nämä tieteiskirjallisuudessa kehitetyt säännöt tekemään robotiikasta turvallista? Miksi? Miksi ei? Mitä muita sääntöjä olisi hyvä olla? Miten nämä säännöt vertautuvat uskontojen ja etiikan normeihin? Ovatko kaikki tähän mennessä noudattaneet näitä sääntöjä?

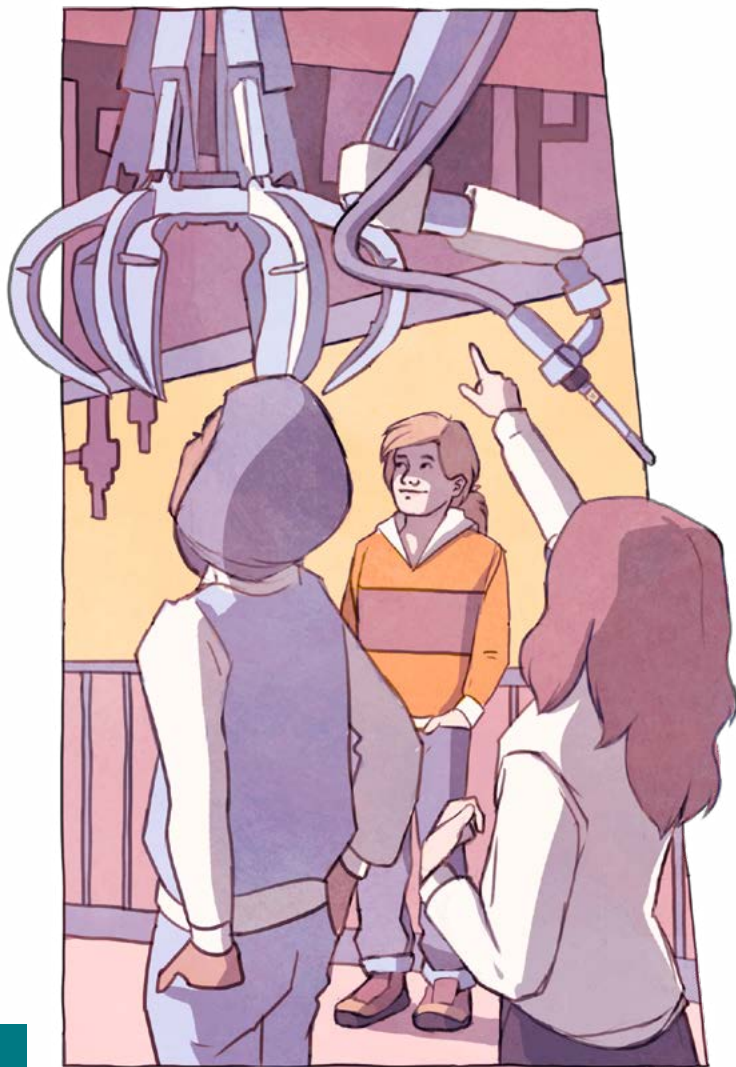
Vinkki!

Voit käyttää seuraavia konkreettisia kysymyksiä keskustelun auttamisessa: Jos tekoälyavustaja tekee antamiesi ohjeiden mukaan puolestasi ostoksia ja ostaakin lainvastaisia tuotteita, kuka on vastuussa? Jos itseohjautuva auto ajaa jalankulkijan päälle, onko vastuussa auton tekoäly, kuljettaja, omistaja, auton valmistaja, tekoälyn ohjelmoija vai joku muu?

Voitte syventää pohdintaa tekemällä pienissä ryhmissä omat perustellut säännöt ja esitellä ne muille ryhmille.



Isaac Asimov on yksi scifi-kirjallisuuden huippukirjailijoista.





28. Video juuri sinulle!

Opitaan, että sisältöpalvelut seuraavat, mitä käyttäjät tekevät ja pyrkivät tarjoamaan lisää kiinnostavaa sisältöä. Kiinnitetään huomiota ja rakennetaan yhdessä ymmärrys palveluiden erilaisista tarkoituksista.

Tutkikaa sisällön suuntaamis- ja suosittelujärjestelmiä, esimerkiksi Yle Areena, YouTube tai muu tuttu videopalvelu. Avatkaa sivusto ja tutkikaa etusivua sekä teille suositeltuja videoita. Ottakaa näyttökuva aloitussivustosta. Käyttäkää viisi minuuttia eri videoiden katsomiseen ja klikkailuun sekä palatkaa etusivulle. Muuttuivatko suositellut videot? Pohtikaa ja keskustelkaa sisällön suuntaamisen hyödyistä ja haitoista. Millä perusteella suosituksia tehdään? Mitä tietoa meistä kerätään ja miksi?



Harjoitus voidaan toteuttaa oppilaiden omilla laitteilla pari- tai pienryhmätyönä tai opettajan johdolla isolla ruudulla. Kannattaa harkita selaimen anonyymitoiminnon käyttöä (incognito window), jolloin selain on ”puhdas”, eikä sisällä käyttäjän vanhoja käyttötietoja. Opettaja voi vaikka avata YouTube:n tällaisessa anonyymi-ikkunassa (jolloin on ensin hyväksyttävä evästeet - joiden säätämismenettelyä voi myös pohtia yhdessä) ja sitten oppilaiden ehdotusten mukaan katsoa pätkät parista videosta. Sitten voidaan palata pääsivulle ja katsoa, miten suositellut videot ovat muuttuneet. Voit ottaa kuvakaappauksen ennen ja jälkeen -tilanteista.

Pohtimalla seurannan syitä päädytään pohtimaan, mikä on palvelun tarkoitus ja mikä taas ei ole sen tarkoitus. Valitkaa jokin video tai sosiaalisen median palvelu sekä pohtikaa sen tarkoitusta ja tavoitteita. Entä mitkä voivat olla käyttäjän tavoitteita? Miten palvelun tarjoajan ja käyttäjän tavoitteet suhteutuvat toisiinsa? Onko palvelun tavoite aina käyttäjän etujen mukainen?



29. Luokan tietokannat

Opitaan tiedon keräämiseen ja käsittelyyn liittyvää kriittisyyttä. Korostetaan henkilötietojen merkitystä ja yksityisyyttä. Tehdään yleistyksiä ja pohditaan yleistysten oikeellisuutta.

Keskustelkaa aluksi yhdessä erilaisista tietokannoista. Millaisia tietokantoja oppilaat tietävät? Mihin heidän tietojaan on kerätty? Entä millaisia tietokantoja omasta luokasta voi muodostaa? Minkälaiset tiedot ovat hyödyllisiä ja mielekkäitä? Mitä tietoja ei saa kerätä yksityisyydensuojan vuoksi? Kerätkää erilaisia tietoja (lempikappale, oikea- tai vasenkätisyys, kynien määrä penaalissa tms.) ja järjestäkää ne tietokannaksi taulukkoon. Käyttäkää esimerkiksi taulukkosovelluksia tai ohjelmointikieltä apuna tiedon keräämisessä ja analysoinnissa.

Varmista opettajana, että kerätyt tiedot eivät ole millään tavalla ahdistavia (esim. henkilön paino); voitte myös kerätä tietoja henkilöiden sijaan jostain ympäristön ilmiöistä.

Löydättekö tietokantaa analysoimalla uutta tietoa tai yllättäviä yhteyksiä (esimerkiksi yhteys värikyynien lukumäärän ja kätisyyden välillä)? Voidaanko yhteyksiä pitää merkittävänä? Mitkä asiat liittyvät toisiinsa, entä mitkä asiat johtuvat toisesta asiasta (korrelaatio vai syy-seuraussuhde)?



30. Tarvitaanko näitä tietoja oikeasti?

Opitaan tarkastelemaan tiedon keräämiseen ja käsittelyyn liittyviä toimintatapoja yksityisyydensuojan näkökulmasta.

Voitte ottaa yksityisyydensuojan ja GDPR-lainsäädännön käsittelyyn konkreettisesti, kun luokka järjestää jonkin tapahtuman, retken tai tekee vaikkapa näytelmän tai muun projektin. Näissä tilanteissa tarvitaan usein luettelo osallistujista ja heidän tehtävistään tai ilmoittautumistietoja tapahtumaan.

Selvitäkää, mitä tietosuoja ja henkilötiedot tarkoittavat. Keskustelkaa yhdessä, mitä henkilötietoja aiotte kerätä tai olette keränneet. Mitä tarkoittaa tietojen anonymisointi ja miksi se on tärkeää? Milloin henkilötiedot tulisi poistaa tietokannasta? Mitä asioita tulee kertoa osallistujille, kun heidän tietojaan lisätään luetteloon esimerkiksi ilmoittautumisen yhteydessä? Mitkä tiedot ovat oikeasti tarpeellisia, mitkä voidaan jättää keräämättä? Mihin tiedot kannattaa tallentaa? Milloin tiedot poistetaan?



GDPR tulee sanoista general data protection regulation (yleinen tietosuoja-asetus). Se on henkilötietojen käsittelyä sääntelevä laki, jota alettiin soveltaa kaikissa EU-maissa keväällä 2018. Tietosuoja-asetus antaa paremman suojan henkilötiedoillesi ja enemmän keinoja hallita tietojesi käsittelyä.

Selkeät perustiedot tietosuojasta löytyvät osoitteesta
<https://tietosuoja.fi/>

Uudet lukutaidot tukena digiajan opetuksessa ja kasvatuksessa

Miksi uudet lukutaidot? Osallisuus ja aktiivinen toimijuus digitaalisessa mediakulttuurissa ja ohjelmoiduissa ympäristöissä edellyttävät monenlaisia taitoja ja osaamista. Nämä taidot ovat jokaiselle kuuluvia kansalaistaitoja. Tutkimukset ja selvitykset koulumaailmasta kuitenkin osoittavat, että oppilaiden digitaalisissa taidoissa ja monilukutaidossa on suuria vaihteluja ja puutteita. Myös opetuskäytännöissä ja opettajien osaamisessa on eroja. Kaikilla tulisi kuitenkin olla tasa-arvoiset mahdollisuudet harjoitella ja oppia digitaalisessa maailmassa tarvittavia taitoja.

Valtakunnallisella Uudet lukutaidot -kehittämishjelmalla (2020–2023) edistetään lasten ja nuorten medialukutaitoa, digitaalista osaamista ja tieto- ja viestintäteknologista osaamista sekä ohjelmointiosaamista varhaiskasvatuksesta esi- ja perusopetukseen. Kehittämishjelman tavoitteena on yhdenvertaisuuden vahvistaminen digitaalisten taitojen opetuksessa ja oppimisessa. Ohjelman on rahoittanut opetus- ja kulttuuriministeriö, ja se on osa laajempaa Oikeus oppia -kehittämishjelmää (2020–2023). Kansallinen audiovisuaalinen instituutti vastaa medialukutaidon ja ohjelmointiosaamisen osakokonaisuuksista ja Opetushallitus digitaalisen osaamisen ja tieto- ja viestintäteknologisen osaamisen osakokonaisuudesta.

Varhaiskasvatussuunnitelman ja esi- ja perusopetuksen opetussuunnitelmien perusteet velvoittavat edistämään lasten ja oppilaiden medialukutaitoa, ohjelmointiosaamista sekä digitaalista osaamista ja tieto- ja viestintäteknologista osaamista niin laaja-alaisena kuin oppimisen alueisiin ja oppiaineisiin kytkeytyvänä osaamisena. Jokaisen kasvatus- ja opetustyössä työskentelevän on omalta osaltaan toteutettava opetussuunnitelmia ja samalla näiden taitojen edistämistä.

Uudet lukutaidot -kehittämishjelman toimenpiteillä tuetaan käytännön kasvatus- ja opetustyön monipuolista ja laadukasta toteuttamista sekä paikallisten suunnitelmien työstämistä ja systemaattista yhteistä kehittämistyötä. Työn tueksi on laadittu kaikkien vapaasti hyödynnettävissä olevia tukimateriaaleja.

Tukimateriaalit löytyvät kehittämishjelman verkkosivuilta www.uudetlukutaidot.fi. Materiaalit julkaistaan myös AOE.fi-palvelussa.

Polkuja

ohjelmointiosaamiseen

Osallisuus ja aktiivinen toimijuus digitaalisessa mediakulttuurissa ja ohjelmoiduissa ympäristöissä edellyttävät monenlaisia taitoja ja osaamista. Nämä taidot ovat jokaiselle kuuluvia kansalaistaitoja.

Kaikilla oppilailla on oikeus saada yhdenvertaisesti opetusta ohjelmointiin sekä hyvään elämään ja aktiiviseen toimintaan ohjelmoiduissa ympäristöissä.

Tämä opas tarjoaa sinulle pedagogisia reittejä ja käytännönläheisiä esimerkkejä ohjelmointiosaamisen edistämiseen yläkouluikäisten nuorten opetuksessa.

Oppaaseen on koottu Uudet lukutaidot -kehittämisohjelmassa julkaistut oppilaan hyvän ohjelmointiosaamisen kuvaukset vuosiluokille 7–9 sekä niiden käytännön toteutusta mallintavia pedagogisia esimerkkejä. Kuvaukset ja pedagogiset esimerkit pohjautuvat perusopetuksen opetussuunnitelman perusteisiin (2014), ja niiden tarkoituksena on tukea ja kannustaa monipuoliseen ohjelmoinnin opetukseen osana eri oppiaineita.

Innostavia oivalluksia ja tekemisen iloa ohjelmointiosaamisen poluille!

Uudet lukutaidot -ohjelma tarjoaa välineitä medialukutaidon, ohjelmointiosaamisen sekä digitaalisen osaamisen ja tieto- ja viestintäteknologisen osaamisen edistämiseen varhaiskasvatuksessa ja esi- ja perusopetuksessa.

www.uudetlukutaidot.fi

UUDET
LUKUTAI
DOT