



# Kuvaohjelmointia Racket-kielellä

Opettajan ohje

**Tiina Partanen**

[RACKET.KOODIAAPINEN.FI](http://RACKET.KOODIAAPINEN.FI)



## Tavoitteet

---

Kuvaohjelmointia Racket – kielellä oppitunneilla opitaan alkeet lausekielisestä ohjelmoinnista. Oppilas tutustuu ohjelmoinnin peruskäsitteisiin kuten funktio, funktion kutsuminen, funktion argumentit, argumenttien tyypit, funktion paluuarvo, paluuarvon tyyppi sekä muuttujan määrittäminen. Kuvien koostaminen yhdistelemällä yksinkertaisia geometrisia muotoja opettaa jakamaan ongelman osiin ja ratkaisemaan osaongelmat erikseen. Omien kuvien ohjelmointi on myös hauskaa, luovaa ja palkitsevan konkreettista.

Oppituntikokonaisuuden jälkeen oppilas osaa käyttää ohjelmointiympäristöä sujuvasti, osaa korjata koodissaan olevia virheitä ja käyttää avukseen Koodarin käsikirjaa.

## Oppiaineet

---

Matematiikka

## Luokkasuositus

---

7. - 9.

Eriyttävänä myös 5. ja 6. luokille

## Kesto

---

3 x 45 min

## Opetussuunnitelma

---

### Laaja-alainen osaaminen

#### Tieto- ja viestintäteknologinen osaaminen (L5)

- Ohjelmointia harjoitellaan osana eri oppiaineiden opintoja.

### Matematiikka

#### Merkitys, arvot ja asenteet

- T1 vahvistaa oppilaan motivaatiota, myönteistä minäkuvaa ja itseluottamusta matematiikan oppijana

#### Työskentelyn taidot

- T5 tukea oppilasta loogista ja luovaa ajattelua vaativien matemaattisten tehtävien ratkaisemisessa ja siinä tarvittavien taitojen kehittämisessä
- T7 rohkaista oppilasta soveltamaan matematiikkaa muissakin oppiaineissa ja ympäröivässä yhteiskunnassa
- T9 opastaa oppilasta soveltamaan tieto- ja viestintäteknologiaa matematiikan opiskelussa sekä ongelmien ratkaisemisessa

#### Käsitteelliset ja tiedonalakohtaiset tavoitteet

- T15 ohjata oppilasta ymmärtämään muuttujan käsite ja tutustuttaa funktion käsitteeseen.
- T16 tukea oppilasta ymmärtämään geometrian käsitteitä ja niiden välisiä yhteyksiä

- T20 ohjata oppilasta kehittämään algoritmista ajatteluaan sekä taitojaan soveltaa matematiikkaa ja ohjelmointia ongelmien ratkaisemiseen

**S1 Ajattelun taidot ja menetelmät:** Harjoitellaan loogista ajattelua vaativia toimintoja kuten sääntöjen ja riippuvuuksien etsimistä ja esittämistä täsmällisesti. Syvennetään algoritmista ajattelua. Ohjelmoidaan ja samalla harjoitellaan hyviä ohjelmointikäytäntöjä. Sovelletaan itse tehtyjä tai valmiita tietokoneohjelmia osana matematiikan opiskelua.

## Oppitunnit

### Välineet

Oppilaat tarvitsevat tietokoneen, tablet-laitteen tai älypuhelimien, jossa on nettiyhteys. Jos laitteita ei ole tarpeeksi tehtävät voidaan tehdä myös parityöskentelynä.

### Ennakkovalmistelut

Koulun pöytäkoneisiin kannattaa asentaa [DrRacket](#)-ohjelma, jos se on mahdollista. Ohjelman avulla oppilaat voivat tallentaa ohjelmakoodin esim. omaan kotihakemistoonsa ja jatkaa sitä seuraavalla tunnilla. Jos ohjelmaa ei voida asentaa tai jos käytetään tablet-laitteita tai älypuhelimia voidaan käyttää myös verkossa toimivaa [WeScheme](#)-sivustoa. WeScheme-sivustolla tehdyt työt voidaan tallentaa google-tunnusten avulla, joten jos oppilailla ei ole käytössä google-tunnuksia, kannattaa luokalle luoda yksi yhteinen [google-tili](#), jotta puolivalmiit ohjelmat saadaan tallennettua ja työtä voidaan jatkaa seuraavalla oppitunnilla.

- ➔ Katso video: [Työkalut DrRacket ja WeScheme](#)

Opettajan kannattaa tehdä harjoitukset ensin itse, että osaa opettaa ne myös oppilailleen.

## 1. Oppitunnin eteneminen

Aluksi kannattaa tutustuttaa oppilaat ohjelmointiympäristön kahteen ikkunaan ja kokeilla yksinkertaisia Racket -kielen laskulausekkeita. Samalla kannattaa selittää Racket-kielen syntaksi ja tutustua virheilmoituksiin.

- ➔ Katso video: [Lausekkeet ja evaluointi](#)
- ➔ Katso opettajan diat: [Peruslaskut](#)

Seuraavaksi tutustutaan kuvakirjaston käyttämiseen Koodarin käsikirjan avulla. Esittele oppilaille miten peruskuvia piirretään.

- ➔ Katso video: [Peruskuviot](#)
- ➔ Katso opettajan diat: [Kuvat \(sivut 1-7\)](#)


Ohjaa oppilaat Racket Koodiaapinen – sivustolle: <http://racket.koodiaapinen.fi>. Ohjeista oppilaat avaamaan oppilaan tehtävät klikkaamalla pinkkiä TEHTÄVÄT logoa:



Avatkaa kuvaharjoitukset kohdasta:

1. Racket alkeet
  - 1.2. Peruskuviot ja kuvien yhdistely
    - 1.2.1. Yksinkertaiset kuvat

← prev up next →



► Oppilaan tehtävät

▼ 1 Racket alkeet

- 1.1 Peruslaskut ja laskujärjestys
- 1.2 Peruskuviot ja kuvien yhdistely
- 1.3 Bonus: Racket Turtle - alkeet

► 1.2 Peruskuviot ja kuvien yhdistely

ON THIS PAGE:

- 1.2.1 Yksinkertaiset kuvat
- 1.2.2 Monimutkaiset kuvat



v.6.4

## 1.2 Peruskuviot ja kuvien yhdistely

Jotta voit piirtää kuvia, ota käyttöön kuvien piirtokirjasto. Kirjoita määrittelyikkunaan:

```
(require 2htdp/image)
```

Voit myös avata tästä tiedoston, jossa tämä vaihe on valmiina:

 Peruskuvat (DrRacket)  
 Peruskuvat (WeScheme)

Ota kirjasto käyttöön painamalla *run*.

Nyt voit tutustua interaktioikkunassa siihen miten peruskuvioita piirretään. Katso käsikirjasta ohjeet ja kokeile näitä:

circle rectangle triangle star ellipse

Opasta oppilaat lataamaan harjoituksen pohjatiedosto (joko DrRacket tai WeScheme). Tallentakaa DrRacket – tiedosto esim. oppilaan kotihakemistoon, muuten muokkaaminen ei ole mahdollista ("File->Save definitions as").

Tämän jälkeen oppilaat voivat tutustua piirtofunktioihin Koodarin käsikirjan avulla (esim. klikkaamalla `ellipse`, avautuu ohje miten `ellipse` - funktiota käytetään jne).

Usein ensimmäisellä tunnilla on käynnistymisvaikeuksia, kaikki oppilaat eivät muista salasanojaan, osaa ladata/tallentaa tiedostoja yms. joten tähän vaiheeseen saattaa kulua tuplasti aikaa. Kaikki on uutta ja ihmeellistä, joten alussa on hyvä edetä hitaasti, että kaikki pysyvät mukana ja saavat onnistumisen kokemuksia.

## 2. Oppitunnin eteneminen

Kun oppilaat ovat tutustuneet ohjelmointiympäristöön ja saaneet aikaiseksi jonkinlaisia kuvioita, on aika puhua funktioista ja niiden argumenteista (muuttujista). Kokeilkaa yhdessä jotakin **funktiota** (esim. `circle`) ja muuttakaa yhtä sen **argumenttia** (esim. sädettä) ja havaitkaa miten muutokset vaikuttavat funktion **paluuarvoon** (ympyrän kuvaan). Viimeistään tässä vaiheessa olisi hyvä puhua myös argumenttien **tietotyypeistä**: säde on tyyppiä **Luku** ja väri on tyyppiä **Merkkijono** (siksi sen ympärillä on sitaatit).

funktion nimi                      argumentit (muuttujat)

```
(circle 100 "solid" "red")
```

Tällä tunnilla opimme nimeämään kuvioita define:n avulla (määrittelemään muuttujia):

- Katso video: [Määrittelyt](#)
- Katso opettajan diat: [Kuvat \(sivut 8-10\)](#)



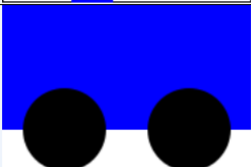
Opimme myös yhdistelemään kuvia uusien funktioiden avulla:

- Katso video: [Kuvien yhdistely](#)
- Katso opettajan diat: [Kuvat \(sivut 11-17\)](#)

Kuvia voidaan yhdistellä myös ilman osien nimeämistä (muuttujien käyttämistä). Nimeäminen helpottaa monimutkaisten kuvien rakentamista ja auttaa jakamaan ongelman osaongelmiin.

Nyt oppilaat ovat valmiita koodaamaan ensimmäiset kuvansa: Japanin lipun ja Suomen lipun tai auton. Oppilaat voivat jatkaa koodaamista samaan tiedostoon kuin edelliselläkin tunnilla tai he voivat aloittaa uuden tiedoston (lataavat tyhjän pohjan kuten 1. tunnilla).

#### Kuvakoodaus 1

Koodaa Japanin lippu.	
Koodaa Suomen lippu.	
Koodaa auto.	

**Vinkki 1:** Oppilaat voivat koodata myös pareittain, ellei työskentely itsenäisesti onnistu. Yksi toimiva tapa on antaa oppilaiden yrittää ratkaista tehtäviä ensin itse, ja jos homma ei aivan kaikilla onnistu voidaan lopuksi koodata kuvia vielä koko luokan voimin yhdessä niin, että yksi oppilas tulee opettajan koneelle ”aivottomaksi” koodausrobotiksi, joka kirjoittaa toisen oppilaan ohjeiden mukaista koodia. Muu luokka seuraa koodausta tykin kautta. Koodausrobotia ja ohjeiden antajaa vaihdetaan tiheään ja jos aikaisemmassa koodissa on ongelmia, uusi pari korjaa ne jne. Näin kaikki oppilaat pääsevät osallistumaan ja asia saattaa avautua paremmin, kun sen näkee isolta ruudulta vielä kertaalleen. Tätä voidaan jatkaa seuraavalla tunnilla niin, että jokainen pääsee vuorollaan molempiin rooleihin.

**Vinkki 2:** Voit katsoa malliratkaisut koodustehtäviin [Racket koodiaapinen -sivuston opettajan materiaalista](#). Materiaali sijaitsee Eliademy – palvelussa ja vaatii tilin avaamisen/kirjautuminen (google tai FB).

## 3. Oppitunti

---

Tällä tunnilla jatketaan edellisen tunnin harjoituksia tai jos ne on jo saatu valmiiksi, voidaan tutustua kuvan muokausfunktioihin.

- Katso video: [Kuvien muokkaus](#)

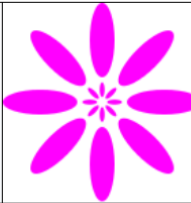


➔ Katso opettajan diat: [Kuvat \(sivut 21-27\)](#)

Avatkaa kuvaharjoitukset kohdasta:

1. Racket alkeet
  - 1.2. Peruskuviot ja kuvien yhdistely
    - 1.2.2. [Monimutkaiset kuvat](#)

Oppilaat voivat jatkaa koodaamista samaan tiedostoon kuin edelliselläkin tunnilla tai he voivat aloittaa uuden tiedoston (lataavat tyhjän pohjan kuten 1. tunnilla).

#### Kuvakoodaus 2

Koodaa kukka.	
Koodaa neliö ympyrän osista.	
Koodaa pikseligrafiikkaa esim. tämä ukkeli tai suunnittele oma pikselikuva!	

**Vinkki 1:** Oppilaat voivat näiden mallikuvien koodaamisen sijaan koodata myös jonkun muun maan lipun tai suunnitella oman logon/pikselikuvan.

**Vinkki 2:** Voit katsoa malliratkaisut koodaustehtäviin [Racket koodiaapinen -sivuston opettajan materiaalista](#). Materiaali sijaitsee Eliademy – palvelussa ja vaatii tilin avaamisen/kirjautumisen (google tai FB).

## Oppituntien arviointi

---

Oppilaan työskentelyä voidaan arvioida aikaansaannosten perusteella. Yksi hyvä tapa tuoda näkyväksi oppilaiden työskentely on tallentaa koodatut kuvat Padlet-seinälle. Näin nähdään millaisia kuvia luokka on koodannut. Opettaja voi tehdä luokalle valmiiksi uuden seinän osoitteessa: <http://padlet.com>. Oppilaat voivat lisätä oman kuvansa seinälle ilman salasanaa, kirjoittamalla osoitteen selaimeen. Kuvan tallentaminen DrRacket:issä tai WeSchemessä onnistuu maalaamalla kuvan hiirellä ja valitsemalla hiiren oikealla painikkeella "Save image as". Kuvan päätteeksi annetaan .png. Tavoitteena ei ole oppia ulkoa ohjelmointikielen komentoja, pääasia on että oppilas saa käsityksen siitä, että ohjelmointi on ongelmanratkaisua, ongelmia voidaan ratkoa pala-palalta ja ohjelmointi voi olla myös luovaa (oman kuvan koodaaminen). Ohjelmointi vaatii tarkkuutta ja kärsivällisyyttä sekä asioiden selville ottamista (tiedon etsintää Koodarin käsikirjasta).

## Lisäopiskelua

---

Racket alkeita voi opiskella myös koodaamalla laskujärjestysharjoituksia.

1. Racket alkeet
  - 1.1. Peruslaskut ja laskujärjestys
    - 1.1.1. [Peruslaskulausekkeita](#)
    - 1.1.2. [Lausekkeita kuvista](#)

## Lisämateriaalia

---

Lisää piirtofunktioita löytyy

Koodarin käsikirjasta (<http://racket.koodiaapinen.fi>) kohdasta:

2. [Kuvat](#)



Oppilaat voivat tutustua mm. siihen mitä värejä on valmiina (3.7. Värit), ja miten niitä tehdään RGB-koodien perusteella (make-color). Myös valmiita kuvatiedostoja voidaan muokata ks. Bitmapit.

## Yhteistyökumppanit

---



Teknologiateollisuuden  
100-VUOTISSÄÄTIÖ