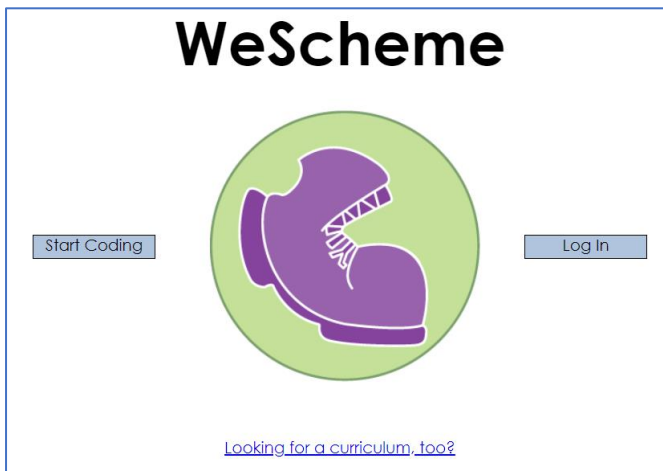


Racket-ohjelmointi



<https://www.wescheme.org/>

Ohjelmakoodin kirjoittamisen voi aloittaa kohdasta **Start Coding**. Jos haluat tallentaa, niin tarvitset Google-tilin.

Ohjelmoinnissa on ideana antaa tarkka toimintaohje, jonka avulla tietokone tekee halutun asian. Toimintaohje eli *algoritmi* täytyy kuvata jollakin tietokoneen ymmärtämällä *ohjelmointikielellä*. Me harjoittelemme Racket-ohjelmointikieltä.

Ohjelmointi vaatii tarkkuutta ja huolellisuutta, koska tietokone noudattaa saamiaan ohjeita mekaanisesti. Pienikin virhe ohjelmassa voi aiheuttaa täysin odottamattoman lopputuloksen. Toisaalta jos ohjelma on laadittu oikein, tietokone suorittaa sen erittäin nopeasti ja virheettömästi.

Ohjelmoinnin oppiminen vaatii paljon aikaa ja hyväksi ohjelmoijaksi kehittyminen vie vuosia aikaa. Ohjelmointia oppii parhaiten käytännön kautta tekemällä paljon pieniä harjoitusohjelmia.

Hyviä ohjeita löytyy lisää Koodarin käsikirjasta: <http://racket.koodiaapinen.fi/manuaali/index.html>

Sisällys

Racket-ohjelmointi.....	1
Peruslaskutoimitukset	2
Lisää laskutoimituksia.....	3
Laskujärjestys.....	4
Satunnaisluku	5
Lausekkeita	6
Merkkijono	7
Ehtolauseet.....	8

Peruslaskutoimitukset

Kaikissa ohjelmointikielissä on omat tarkat säännöt siitä, miten esimerkiksi peruslaskutoimitukset kirjoitetaan komennoissa. Näitä kirjoitussääntöjä kutsutaan nimellä **syntaksi**.

Aritmeettisiä operaattoreita ovat muunmuassa

+	summa	expt	potenssi
-	erotus	sqrt	neliöjuuri
*	tulo	abs	itseisarvo
/	osamäärä		

Racket-kielessä laskulausekkeet kirjoitetaan sulkujen sisään. Laskutoimitusmerkki kirjoitetaan sulkujen sisään ensimmäisenä. Tämän jälkeen luetellaan luvut peräkkäin niin, että etumerkki on kiinni luvussa. Välilyöntejä lukujen välistä ei voi jättää pois

INTERAKTIOIKKUNA

Peruslaskutoimitukset

Enter

välilyönti

28 + 7

(+ 28 7)
35

28 - 7

(- 28 7)
21

28 · 7

(* 28 7)
196


28 : 7

(/ 28 7)
4



Laskutoimitus kertoo, mikä on kutsuttava toiminto eli **funktion nimi**. Laskulausekkeen lukuja kutsutaan funktion **argumenteiksi**.

Laske lausekkeen arvo päässä. Tarkista vastaus ohjelmoimalla.

Run 

Tehtävä 1

a) $9 + 10$

b) $7 + 6$

c) $3 + 18$

Tehtävä 2

a) $16 - 2$

b) $42 - 10$

c) $71 - 9$

Tehtävä 3

a) $4 \cdot 9$

b) $20 \cdot 6$

c) $3 \cdot 15$

Tehtävä 4

a) $16 : 8$

b) $49 : 7$

c) $18 : 2$

Lisää laskutoimituksia

Laskutoimituksia

potenssi

(expt 2 5)

32

neliöjuuri

(sqrt 49)

7

itseisarvo

(abs -10)

10

Laske lausekkeen arvo.

Tehtävä 5

a) 7^2

b) 9^3

c) $(-2)^5$

d) $(1 - 5)^4$

Tehtävä 6

a) $\sqrt{9}$

b) $\sqrt{169}$

c) $-\sqrt{4}$

d) $\sqrt{36 + 64}$

Tehtävä 7

a) $|-6|$

b) $|-0|$

c) $|-8 - 1|$

d) $4 \cdot |-7|$

Laskujärjestys

Matemaattisen lausekkeen laskujärjestys voidaan kirjoittaa Racket-kielessä sisäkkäisten komentojen avulla. Jokainen yksittäinen laskutoimitus alkaa Racket-komennossa sululla ja päättyy sulkuun. Suoritusjärjestys alkaa sisimmäisistä suluista. Rinnakkaiset sulkumerkeillä merkityt komennot suoritetaan järjestyksessä vasemmalta oikealle.

Esimerkki 1 Summalauseke $9 + 5 \cdot 3$ kirjoitetaan komentona

`(+ 9 (* 5 3))`

Esimerkki 2 Osamäärä $(100 - 20) : (16 + 4)$ kirjoitetaan komentona

`(/ (- 100 20) (+ 16 4))`

Esimerkki 3 Tulo $3 \cdot [8 + 7 \cdot (-10)]$ kirjoitetaan komentona

`(* 3 (+ 8 (* 7 -10)))`

Kirjoita ohjelmakoodit, jotka laskevat seuraavat laskut

Laske lausekkeen arvo.

Tehtävä 8


a) $4 + 8 + 3$

b) $4 - 8 + 3$

Tehtävä 9

a) $(1 - 6) \cdot 2$

b) $1 - 6 \cdot 2$

Run 

Satunnaisluku

MÄÄRITTELYIKKUNA

Satunnaislukuja

luku 0, 1, 2 tai 3 (random 4)
3

luku 0, 10 tai 20 (* (random 3) 10)
20

Run ▶

Satunnaisluku on satunnaisesti arvottu kokonaisluku, joka ei ole negatiivinen ja jonka on pienempi kuin annettu yläraja.

Tehtävä 10

Mitä satunnaislukuja komento arpoo?

- a) (random 6) b) (random 2) c) (random 11)

Tehtävä 11

Arvo satunnaisluku, joka on pienempi kuin

- a) 8 b) 12 c) 100 d) 7 500

Lausekkeita

Matematiikassa muuttujaa merkitään kirjaimella. Usein käytetään kirjaimia x , y , a ja b . Kirjaimella merkitylle muuttujalle voidaan antaa laskussa vakiona pysyvä arvo, joka ei muutu matematiikan laskun tai tietokoneohjelman suorituksen aikana. Lauseke on laskutoimitus, joka on kirjoitettu muuttujien ja lukujen avulla. Tietyn muuttujan arvon avulla voidaan laskea lausekkeen arvo.

Jos muuttujalle annetaan ohjelmassa tietty vakiona pysyvä arvo, se tarkoittaa, että sitä ei voi muuttaa tietokoneohjelman suorituksen aikana. Racket-kielessä ohjelman aikainen **vakio** määritetään **define**-komennolla, jossa yksittäisellä kirjaimella tai sanalla merkitylle muuttujalle annetaan arvo. Matematiikassa käytetään yleensä yksittäisiä muuttujakirjaimia. Muuttujat kannattaa **nimetä** järkevästi. Ohjelmoinnissa käytetään sekä yksittäisiä kirjaimia että pitempiä, käyttötarkoitusta kuvaavia nimiä. Näissä nimissä ei saa olla välilyöntejä. Myöskään Racket-kielen komentoja ei voi käyttää tarkoittamaan jotain muuta. Nimessä ei myöskään saa olla sulku- tai heittomerkkejä eikä merkkejä ' ` | ; # .

Esimerkki 1 Laske lausekkeen $3a + 6$ arvo, kun $a = 4$. Sijoittamalla luku 4 muuttujan a paikalle saadaan

$$3 \cdot 4 + 6 = 12 + 6 = 18.$$

Yllä oleva lasku voidaan kirjoittaa Racket-kielillä komentoina:

```
(define a 4)
(+ (* 3 a) 6)
```

Tehtävä 12

Kirjoita yllä olevat koodirivit määrittelyikkunaan ja suorita komennot. Tuliko oikea lausekkeen arvo? Kokeile antaa a :lle eri arvoja.

Tehtävä 13

Laske lausekkeen arvo, kun $x = -5$.

- a) $x + 7$ b) $x - 9$ c) $11x$ d) $-x + 3$

Tehtävä 14

Laske lausekkeen arvo, kun $a = 7$.

- a) $a - 9$ b) $16a$ c) $2a + 8$ d) $-3a - 11$

Merkkijono

Merkkijono on tekstiä, joka koostuu teksti-, numero- ja erikoismerkeistä sekä välilyönneistä. Kirjaimista muodostetaan sanoja ja numeroista lukuja. Erikoismerkkejä ovat esimerkiksi pilkku, piste, tavuviiva, kauttaviiva, kysymys- ja huutomerkki.

: # % & / ; + - ! ? _ - *

Racket-kielen merkkijonossa ei saa käyttää heittomerkkiä " ", sillä heittomerkkien avulla merkitään merkkijonon alkaminen ja loppuminen. Myöskään sulkumerkit eivät kelpaa osaksi merkkijonoa. Racket-kielen **text-komento** muuntaa merkkijonon kuvaksi.

Text-komennossa ensimmäisenä sulun jälkeen kirjoitetaan haluttu teksti (merkkijono) lainausmerkkien " " sisään. Tämän jälkeen kerrotaan fontin koko lukuna sekä väri tekstinä heittomerkkien sisällä. Viimeisenä komennossa on sulkumerkki. Racket-kielessä on käytössä olevat Värit löytyvät [värilistasta](#), jossa on valmiina värien nimet heittomerkkien sisällä.

Perusvärejä: "red", "white", "black", "yellow", "green", "blue", "gray"

Merkkijono

Merkkijono (string, engl.) on tekstiä, joka koostuu teksti-, numero- ja erikoismerkeistä. Heittomerkkien sisällä oleva teksti tulostuu sellaisenaan.



Tehtävä 15

Kirjoita alla olevat komennot siten, että viivojen tilalle tulee omat tietosi.

(text "Hei! Minun nimeni on _____ ." 20 "violet")

(text "Olen ___-luokalla." 20 "royalblue")

Vaihda tekstiä, tekstin kokoa ja värejä. Tutki [värilistan](#) värejä.

Tehtävä 16

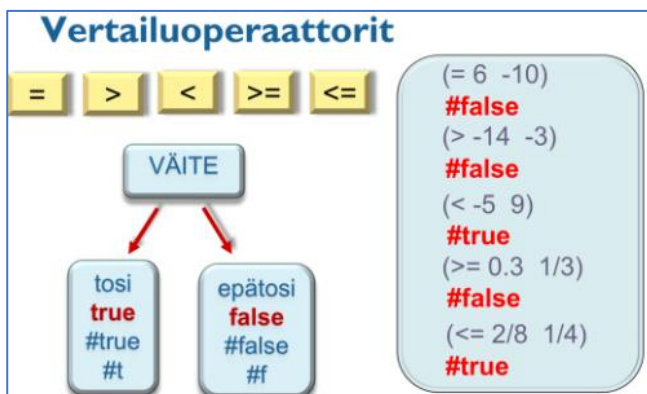
Kirjoita text-komennolla x-kirjaimia niin, että niistä muodostuu jokin kuvio (esim. alla oleva).

```
XXXXXXXXXXXXXXXXXX
XXX             XXX
XXX             XXX
XXX             XXX
XXX             XXX
XXX             XXX
XXX             XXX
XXX             XXX
XXXXXXXXXXXXXXXXXX
```

Ehtolauseet

Lukuja tai lausekkeiden arvoja voi verrata matematiikasta tutuilla merkinnöillä.

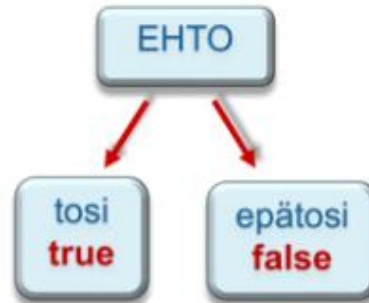
Merkintä	Racket	Merkitys
=	=	yhtäsuuri kuin
>	>	suurempi kuin
<	<	pienempi kuin
≥	>=	suurempi tai yhtäsuuri kuin
≤	<=	pienempi tai yhtäsuuri kuin



Ehtorakenne

Boolean ehdon totuusarvo on joko tosi tai epätosi.

(if (**Boolean ehto**)
tosi-haara
epätosi-haara)



Kirjoita alla oleva esimerkki ja testaa koodin toimivuutta kirjoittamalla esim. käsky (luku 9)

```

(define (luku x)
  (if (number? x)
      (if (= x 0)
          (text "Luku on nolla." 20 "black")
          (if (> x 0)
              (text "Luku on positiivinen." 20 "red")
              (text "Luku on negatiivinen." 20 "blue")))
      (text "Et kirjoittanut lukua." 20 "darkgreen")))
  
```

Tehtävä 17

Kirjoita funktio, joka vertaa annettua lukua lukuun 100.

```

> (vertaa 4)
Luku on pienempi kuin 100.
> (vertaa 170)
Luku on vähintään 100.
  
```

Tehtävä 18

Kirjoita funktio, joka tarkistaa pin-koodin.

```

> (pin 6500)
Kirjoittamasi koodi on väärin!
> (pin 8923)
Kirjoittamasi koodi on oikein!
  
```

Tehtävä 19

a) Kirjoita funktio, joka vertaa käytössä olevaa rahan määrää ostoksen hintaan.

```

> (ostos 45 30)
Voit tehdä ostoksen.
> (ostos 75 100)
Rahaa on liian vähän.
  
```

b) Lisää ohjelmaan virheilmoitus.

```

> (ostos -17 32)
Tiedot ovat väärinä.
  
```

Tehtävä 20

Kirjoita funktio, joka nimeää kulman sen asteluvun mukaan.

nolla-kulma	suora-kulma	oiko-kulma	täysi-kulma
0°	90°	180°	360°

terävä kulma	tylppä kulma	kupera kulma
0° < α < 90°	90° < α < 180°	180° < α < 360°