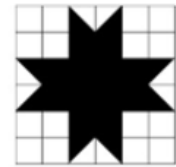
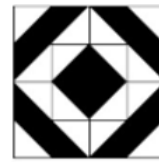


## Kuvien piirtäminen ohjelmoimalla



## Sisällys

Kuvien piirtäminen ohjelmoimalla .....	1
Kuvaohjelmointia.....	1
Tehtävät.....	3
Kuvioiden yhdistäminen ( <b>Beside ja above</b> ).....	5
Tehtävät.....	6
Symmetriset kuviot ( <b>flip-vertical ja flip-horizontal</b> ) .....	8
Tehtävät.....	10
Ekstra tehtävät .....	10

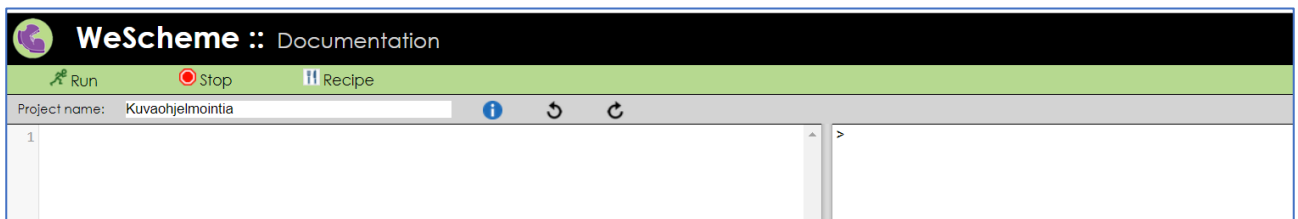
Ohjelmointi perustuu funktioiden käyttämiseen. Funktion koodiin on tallennettu yleiset toimintaohjeet, joita seuraamalla tietokoneohjelma pystyy tuottamaan halutunlaisen lopputuloksen. Jotta jokaiselle vähän erilaiselle tehtävälle ei tarvitse kirjoittaa omaa ohjetta, funktiot kirjoitetaan niin, että ne pystyvät tekemään samantyyppisiä asioita hieman eritavoin esim. yksi funktio pystyy piirtämään eri kokoisia ja eri värisiä ympyröitä. Tämä saadaan aikaan kirjoittamalla toimintaohjeet joustaviksi muuttujien avulla. Muuttujia käytetään kuvaamaan asioita, joita halutaan vaihdella, esim. ympyrän säde tai väri voisivat olla muuttujia.

Nyt tutustutaan kuvien piirtämiseen funktioiden avulla. Ohjeet (funktiot) ovat valmiina, sinä päätät mitä muuttujien arvoja käytetään.

## Kuvaohjelmointia

**Lue esimerkit. Kirjoita esimerkkien koodirivit Weschemeen. Aloita avaamalla alla olevan linkin harjoitustiedosto ja paina Edit .**

<https://www.wescheme.org/view?publicId=dZVEWNWswE>



## 0) Kuvakirjasto

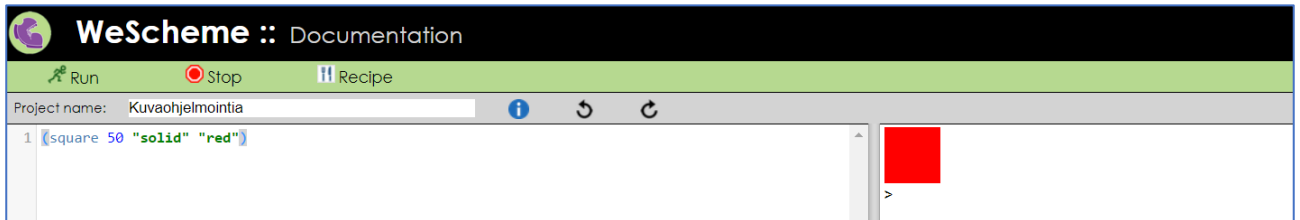
Testaa näin:

- Kirjoita interaktioikkunaan:  

```
(square 50 "solid" "red")
```

Kirjoita täsmälleen samalla tavalla kuin mallissa (sulut, välit ja heittomerkit)

- **Paina Run**



## 1) square

a) Voit piirtää neliöitä `square` -funktion avulla. Annettu luku kertoo neliön sivun pituuden pikseleissä. Kokeile mitä seuraavat koodirivit tekevät:

```
(square 100 "solid" "green")  
(square 150 "outline" "blue")
```

Lisää värejä löytyy Koodarin käsikirjasta

<http://racket.koodiaapinen.fi/manuaali/varilista.html>

## b) Tutki

- Mitä tapahtuu, jos annat sivun pituudeksi nollan? Entä jos se on negatiivinen?
- Mitä "solid" ja "outline" tarkoittavat? Mitä tapahtuu, jos annat näiden sijaan kokonaisluvun väliltä 0-255?
- Miten saat piirrettyä keltaisia, violetteja, mustia, ruskeita tai oransseja neliöitä?

## 2) rectangle

a) Voit piirtää suorakulmioita `rectangle` -funktion avulla. Kokeile:


```
(rectangle 100 50 "solid" "red")
```

b) Tutki

- Mitä luvut 100 ja 50 tarkoittavat?
- Miksi tarvitaan kaksi lukua?
- Onko lukujen järjestyksellä väliä? Kokeile vaihtaa lukujen järjestystä.

### Tehtävät

Piirrä ohjeen mukaiset kuvat

Run 

1. Piirrä viiden suorakulmion sarja, jossa muutat vain yhtä ominaisuutta kerrallaan (saat aikaan viisi riviä koodia).  
Lähde liikkeelle tästä koodista (1. rivi):  

```
(rectangle 100 50 "solid" "red")
```
2. Piirrä neljän suorakulmion sarja, jossa kannan ja korkeuden suhteet ovat:  
a) 3:5    b) 4:1    c) 25:9    d) 20:3

*Vinkki: skaalaa luvut niin, että saat suorakulmioiden sarjan mahtumaan ruudulle*

3. Piirrä 4 keskenään erimuotoista suorakulmiota, joiden jokaisen piiri on 400 (piirrä vain reunaviiva). Laske<sup>\*)</sup> syntyneiden suorakulmioiden pinta-alat ja järjestä koodirivit niin, että suorakulmioiden kuvat tulostuvat pinta-alan mukaan pienimmästä suurimpaan. Voit merkitä laskemasi pinta-alan koodirivin loppuun kommentiksi erottamalla sen koodista puolipisteen ; avulla:  

```
(rectangle 100 100 "outline" "red"); A=10000
```

Kommentti

<sup>\*)</sup> Voit käyttää interaktioikkunaa apuna, ja laskea pinta-alan sen avulla. Esim. kirjoita:  

```
(* 100 100)
```

 ja paina "enter".

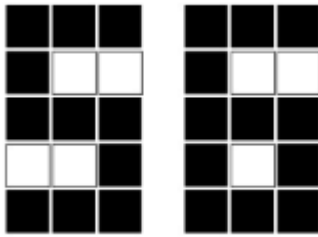
4. Piirrä 4 keskenään erimuotoista suorakulmiota, joiden jokaisen pinta-ala on 90 000 (väritä kuviot sisältä). Laske syntyneiden suorakulmioiden piirit ja järjestä koodirivit niin, että suorakulmioiden kuvat tulostuvat piirin pituuden mukaan pienimmästä suurimpaan. Voit merkitä laskemasi piirin pituuden koodirivin loppuun kommentiksi erottamalla sen koodista puolipisteen ; avulla:

```
(rectangle 300 300 "solid" "blue") ; p=1200
```

Kommentti

## Kuvioiden yhdistäminen (Beside ja above)

Isot kuvat muodostuvat pienistä kuvista



Lue esimerkit. Kirjoita koodirivit Weschemeen. Avaa harjoitustiedosto ja paina Edit

<https://www.wescheme.org/view?publicId=t2JqVZEgbP>

### Esivalmistelut

#### 0) Valmiit kuvat

Sinulla on käytössä seuraavat peruskuviot:

K1:  K2:  K3:  K4: 

Testaa kuvia näin:

- Kirjoita interaktioikkunaan K1
- Saitko toimimaan?

Huom!

Kirjoita iso K-kirjain

### Tutustutaan funktioihin

#### 1) beside

a) Yhdistä kuvioita vierekkäin `beside` -funktion avulla. Kokeile mitä seuraava koodirivi tekee:

```
(beside K1 K2)
```

b) Entä jos vaihdat K1:n ja K2:n järjestystä?

```
(beside K2 K1)
```

c) Kokeile, voitko yhdistää kolme kuvaa?

## 2) above

a) Yhdistä kuvioita allekkain `above` -funktion avulla. Kokeile:

```
(above K1 K3)
```

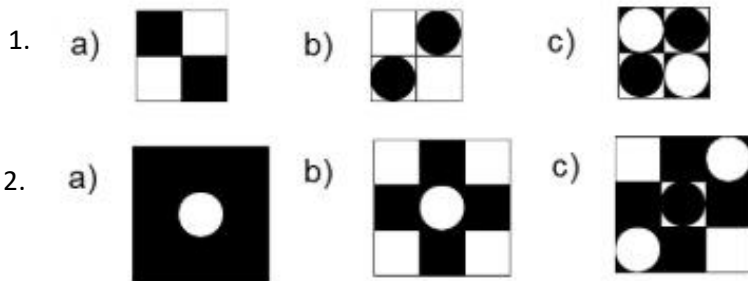
b) Voit yhdistää `beside`:n ja `above`:n samaan lausekkeeseen. Kokeile:

```
(beside (above K1 K3)
         (above K3 K1))
```

## Tehtävät

Piirrä seuraavat kuvat:

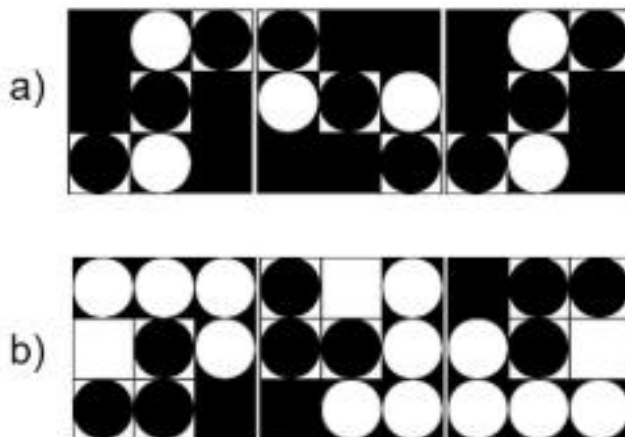
*(vinkki: above, beside, K1, K2, K3, K4)*



Keksi oma kuvio, suunnittele se ensin ruutupaperille, ja koodaa se sitten.

*(vinkki: kokeile vaihtaa musta joksikin toiseksi väriksi, keksitkö miten).*

3. Jatka kuvasarjaa ja koodaa puuttuva kuva:



4. Hassut otukset 1

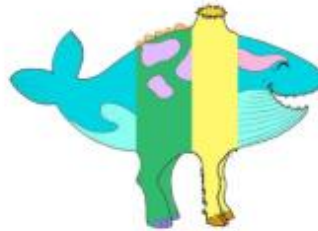
Mitä jos eläimet voisivat vaihtaa osia keskenään?



Montako erilaista eläintä voit koota, kun sinulla on kolmeen osaan jaetut valas, kameli ja dino?  
Koodaa kaikki vaihtoehdot. Tiedostosta löytyvät valmiit eläinpalat (kameli: K1, K2, K3, dino: D1, D2, D3 ja valas: V1, V2, V3)

5. Hassut otukset 2

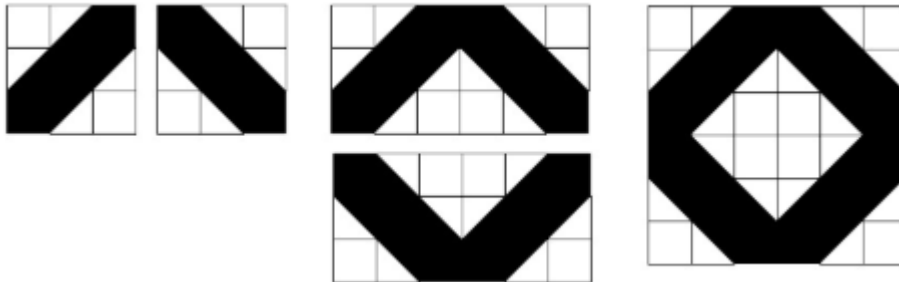
Mitä jos eläimet voisivat vaihtaa osia keskenään?



Montako erilaista eläintä voit koota, kun sinulla on neljään osaan jaetut valas, kameli ja dino?  
Tiedostosta löytyvät valmiit eläinpalat (kameli: K1, K2a, K2b, K3, dino: D1, D2a, D2b, D3 ja valas: V1, V2a, V2b, V3). Kokoa mielestäsi hassuin mahdollinen eläin.

## Symmetriset kuviot (flip-vertical ja flip-horizontal)

Symmetriset kuvat koostuvat osasten peilikuvista






Lue esimerkit. Kirjoita koodirivit Weschemeen. Avaa harjoitustiedosto ja paina Edit

<https://www.wescheme.org/view?publicId=azUoD9LrAY>

### Esivalmistelut

#### 0) Valmiit kuvat

Sinulla on käytössä seuraavat peruskuviot:

K1:  K2:  K5: 

Testaa kuvia näin:

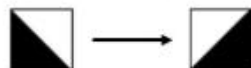
- Kirjoita interaktioikkunaan K5
- Saitko toimimaan?

### Tutustutaan funktioihin

#### 1) flip-horizontal

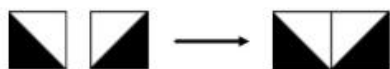
a) Peilaa mikä tahansa valmis kuva vaakasuunnassa `flip-horizontal` -funktion avulla. Kokeile<sup>\*)</sup> mitä seuraavat koodirivit tekevät:

```
K5  
(flip-horizontal K5)
```



**b)** Voit yhdistää syntyneen kuvan alkuperäisen kuvan kanssa. Kokeile:

```
(beside K5 (flip-horizontal K5))
```



**c)** Kopioi edellinen koodi määrittelyikkunaan. Anna uudelle palalle nimi `define:n` avulla ja paina "run".

```
(define OSA1  
  (beside K5 (flip-horizontal K5)))
```

**d)** Testaa, että pala toimii oikein. Kirjoita interaktioikkunaan `OSA1`



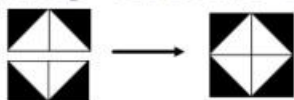
## 2) flip-vertical

**a)** Voit peilata kuvan myös pystysuunnassa `flip-vertical` -funktion avulla. Kokeile:

```
(flip-vertical OSA1)
```






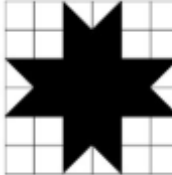
**b)** Voit nyt yhdistää syntyneen kuvan alkuperäisen kuvan kanssa. Kokeile:

```
(above (flip-vertical OSA1) OSA1)
```



## Tehtävät

Piirrä seuraavat kuvat: *(vinkki: flip-vertical, flip-horizontal, beside, above, K5)*

- a)  b)  c) 
- a)  b)  c) 
- Keksi oma kuvio, suunnittele se ensin ruutupaperille, ja koodaa se sitten.

## Ekstra tehtävät

\* Harjoitus: Kuvakoodaus

\* Harjoitus: Kuvioita

\* Harjoitus: Turtle-grafiikkaa 1

\* Harjoitus: Turtle-grafiikkaa 2

\* Harjoitus: Turtle-grafiikkaa 3

\* Harjoitus: Turle-grafiikkaa 4

\*Harjoitus: Frozen <https://studio.code.org/s/frozen/stage/1/puzzle/1>

\*Harjoitus: Taiteilija <https://studio.code.org/s/artist/stage/1/puzzle/1>