

3D KAPPALEITA OHJELMOIMALLA <https://editor.p5js.org/>

Tällä tunnilla piirrämme 3D kappaleita ohjelmoimalla. Ohjelmointiympäristönä meillä on p5.js, joka on JavaScript-pohjainen ohjelmointiympäristö, joka toimii täysin selaimella eli ohjelmointi onnistuu millä tahansa laitteella (tietokone, tabletti, ChromeBook), jolla pääsee Internetiin. Ohjelmointiympäristö löytyy hakusanalla: "p5js editor" tai voi mennä suoraan yllä olevaan osoitteeseen. Suosittelen, että luot itsellesi tunnuksen ohjelmointiympäristöön, niin voit tallentaa kirjoittamasi koodin muistiin.

3D OHJELMAN RUNKO

Katsotaan aluksi 3D ohjelman runkoa eli ohjelman tarjoamaa valmista koodia

```
function setup() {
  createCanvas(800, 800, WEBGL);
}

function draw() {
  background(0);
  // Tämä on kommentti
}
```

Ohjelma koostuu kahdesta osasta: kahdesta lohkoista tai kahdesta funktiosta, jotka ovat nimeltään setup- ja draw-lohko. Kummassakin lohkoissa on ensin kaarisulkeet (), siksi että sulkuihin voitaisiin laittaa funktion sisälle viettäviä arvoja. Koska mitään välitettäviä arvoja ei ole, niin sulkujen sisälle ei laiteta mitään. Sen jälkeen on aaltosulkeet { }, joka tarkoittaa lohkon aloitus- ja lopetus merkkiä. Kaikki ohjelman tarvitsemat komennot yleensä kirjoitetaan aaltosulkeiden väliin.

Setup-lohko suoritetaan ensin ja se suoritetaan vain kerran. Setup-lohkoissa on yksi pakollinen komento: createCanvas(800,800, WEBGL); jolla määritellään ikkunan ja samalla käytettävissä olevan xy-koordinaatiston koko. Voit aivan vapaasti muuttaa lukujen 800 paikalle jonkin toisen luvun eli muuttaa ikkunan kokoa. Lisämääre WEBGL ottaa käyttöön xyz -koordinaatiston. Itse komento: createCanvas, on kirjoitettava täysin samalla tavalla eli pienillä kirjaimilla yhteen, paitsi Canvas-sana, joka alkaa isolla.

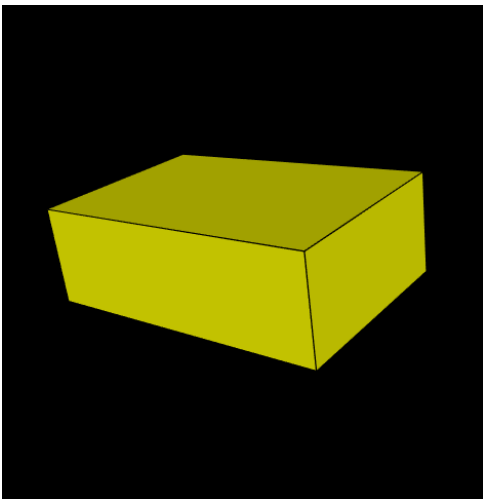
Draw-lohko on ns. pääohjelma, joka on ikuisessa silmukassa. Laitamme kaikki omat komennot draw -lohkoon aaltosulkeiden { ja } väliin. Laita yksi komento yhdelle riville. Huomaa, että jokainen komento loppuu puolipisteeseen. Komentojen lisäksi ohjelmaan voimme lisätä kommentteja. Kaikki kommentit alkavat kahdella kauttaviivalla: //. Kommentit eivät vaikuta ohjelma suoritukseen millään tavalla. Draw -lohkon alkuun on hyvä lisätä komento: background(0); joka tyhjentää ikkunan ja samalla värittää ikkunan mustaksi. Jos muutat lukuarvon 0 tilalle 255, niin ikkuna väritetään valkeaksi. Pienin arvo on 0 ja suurin 255.

SUORAKULMAISEN SÄRMIÖN PIIRTÄMINEN

Suorakulmainen särmiö piirretään komennolla: **box(pituus, leveys, korkeus)**; Tehdään ohjelma, joka piirtää keltaisen suorakulmaisen särmiön, jonka pituus on 100, leveys on 200 ja korkeus on 300. Lisätään ohjelmaan kommentit.

```
function setup() {
  createCanvas(800, 800, WEBGL); // Ikkunan koko 800 x 800
}

function draw() {
  background(0); // Taustaväri musta
  fill(255,255,0); // Täyttöväri keltainen
  lights(); // Luodaan valot ja varjot
  rotateZ(radians(60)); // Käännä z akselia 60 astetta
  rotateX(radians(45)); // Käännä x akselia 45 astetta
  rotateY(radians(30)); // Käännä y akselia 30 astetta
  box(100,200,300); // Piirrä suorakulmainen särmiö
}
```



Suorakulmaista kolmiota voidaan pyörittää eri asentoihin rotateX(kulma1), rotateY(kulma2) ja rotateZ(kulma3) komennoilla. Tässä kulma on annettava radiaani-asteina. Laittamalla lisäkommento: radians(kulma) voidaan asteet antaa tavallisina asteina, jolloin ohjelma muuttaa automaattisesti kulmat radiaaniasteiksi.

Kokeile muuttaa yllä olevan ohjelman lukuarvoja niin box-komennoissa, kuin rotate-komennoissa.

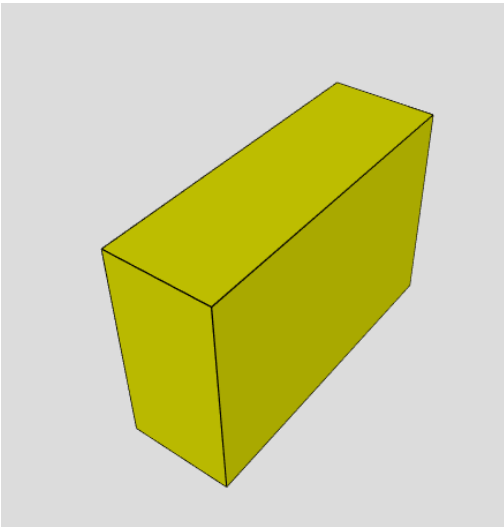
KUVION PYÖRITTÄMINEN HIIRELLÄ

```
function setup() {
  createCanvas(800, 800, WEBGL);
}

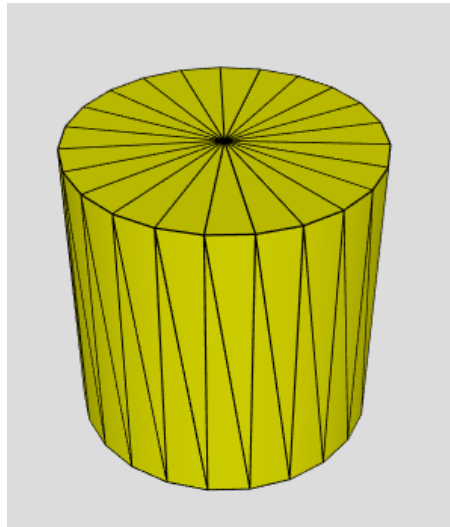
function draw() {
  background(0); // Taustaväri musta
  fill(255,255,0); // Täyttöväri keltainen
  lights(); // Luodaan valot ja varjot
  rotateX(radians(-mouseY)); // Pyöritä x-akselin ympäri
  rotateY(radians(mouseX)); // Pyöritä y-akselin ympäri
  box(100, 200, 300); // Piirrä suorakulmainen särmiö
}
```

Kun lisää lukuarvon tilalle rotate-komennon sisälle: mouseX tai mouseY komennon, niin kuvio saadaan pyörimään kolmiulotteisesti hiiren avulla.

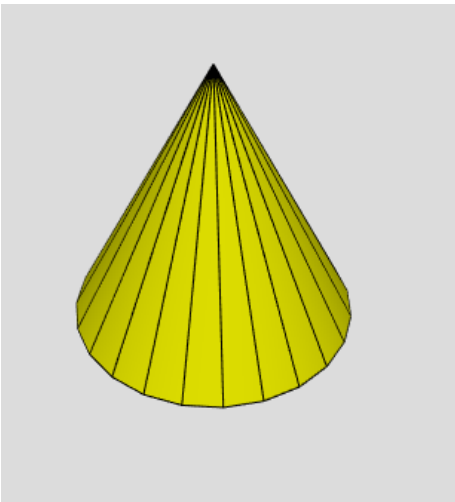
3D KAPPALEET P5.JS-OHJELMOINNISSA



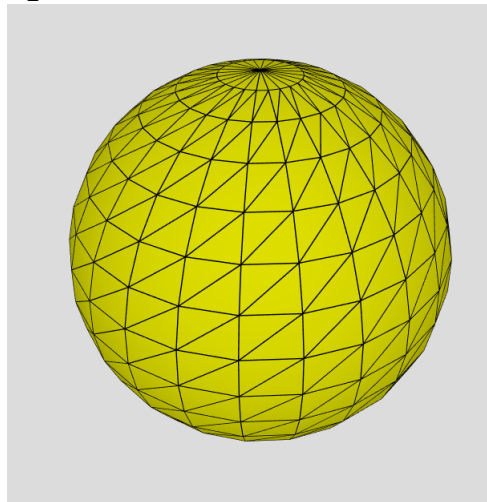
`box(pituus, leveys, korkeus);`



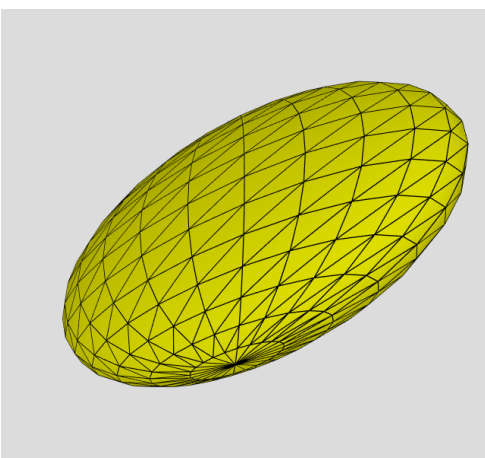
`cylinder(säde, korkeus);`
`cylinder(säde, korkeus, tahkot);`



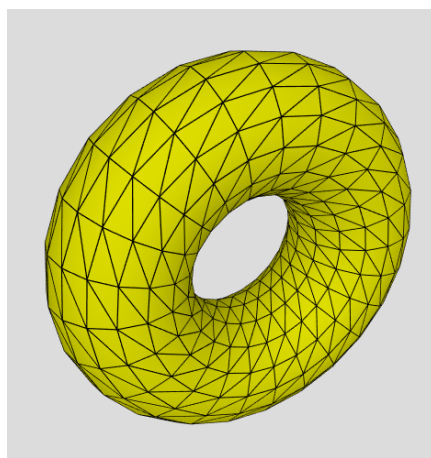
`cone(säde, korkeus);`
`cone(säde, korkeus, tahkot);`



`sphere(säde);`



`ellipsoid(pituus, leveys, korkeus);`



`torus(säde1, säde2);`















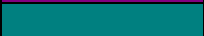
Kootaan vielä taulukkoon tarvittavat piirtokomennot

KOMENTO	MERKITYS
box(pituus, leveys, korkeus);	box() piirtää suorakulmaisen särmiön pituuden, leveyden ja korkeuden avulla.
cylinder(säde, korkeus); cylinder(säde, korkeus, tahkot);	cylinder() piirtää ympyräpohjaisen lieriön pohjaympyrän säteen ja lieriön korkeuden avulla.
cone(säde, korkeus); cone(säde, korkeus, tahkot);	cone() piirtää ympyräpohjaisen kartion pohjaympyrän säteen ja kartion korkeuden avulla.
sphere(säde);	sphere(); piirtää pallon säteen avulla.
ellipsoid(säde1, säde2, säde3);	ellipsoid() piirtää ellipsoidin eli litistetyn pallon jokaisella xyz-akselilla olevan eri säteen avulla.
torus(säde1, säde2);	Geometrinen pyörähdyspinta, joka syntyy kun ympyrä, jonka säde on säde2 pyörähtää toisen säteen säde1 ympäri.

Huomio: cylinder- ja cone-komennolla voidaan myös piirtää särmiö tai pyramidi, jossa pohja on monikulmio, kun lisätään vielä kolmas arvo, joka määrää tahkojen kokonaismäärän. Kokeile esimerkiksi laittaa cone(100,200); komentoon kolmas luku: cone(100,200,5); Tässä lukuarvo 5 tarkoittaa sivutahkojen lukumäärää, jolloin kuviosta syntyy pyramidi, jossa pohja on neliö. Pohja lasketaan myös yhdeksi tahkoksi. Suurin mahdollinen arvo tahkojen määrälle on kuitenkin 24.

RGB-VÄRIT

Katsotaan vielä kuinka muuttelemalla lukuarvoja fill()-komennossa saadaan aikaiseksi eri värejä.

R	G	B	KOMENTO	NIMI	VÄRI
255	0	0	fill(255,0,0);	Punainen	
0	255	0	fill(0,255,0);	Vihreä, lime	
0	0	255	fill(0,0,255);	Sininen	
255	255	0	fill(255,255,0);	Keltainen	
255	0	255	fill(255,0,255);	Magneta, fuksia	
0	255	255	fill(0,255,255);	Syaani	
0	0	0	fill(0,0,0);	Musta	
255	255	255	fill(255,255,255);	Valkea	
128	128	128	fill(128,128,128);	Harmaa	
128	0	0	fill(128,0,0);	Bordeaux	
0	128	0	fill(0,128,0);	Vihreä	
0	0	128	fill(0,0,128);	Laivaston sininen	
128	128	0	fill(128,128,0);	Oliivinvihreä	
128	0	128	fill(128,0,128);	Purppura	
0	128	128	fill(0,128,128);	Sinivihreä, teal	

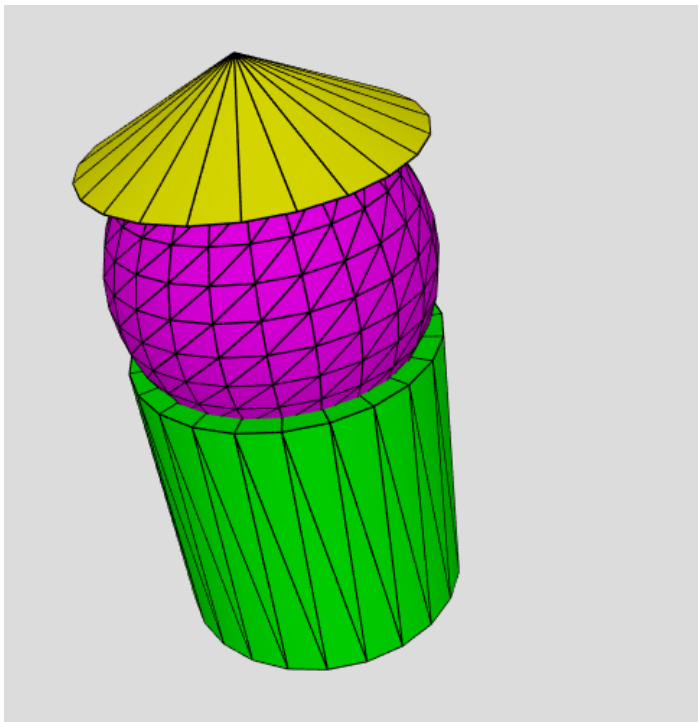
Vinkki: Voit lisätä fill(); komentoon neljännen lukuarvon, jonka avulla voit tehdä kuvioista läpinäkyviä. Läpinäkyvyys arvolla pienin arvo on 0 (= täysin läpinäkyvä) ja suurin arvo 255 (= täysin peittävä). Oletus on arvo 255 eli läpinäkyvyys ei ole tällöin käytössä. Esimerkiksi voit kokeilla muuttaa fill-komennon muotoon: `fill(255,255,0,150);`

YHDISTELMÄKUVIOT

Kaikki kuviot piirretään aina origoon eli komennoissa ei ole kuvion xyz- paikkaa ilmaisevaa arvoa. Kun haluat piirtää useita erilaisia kuvioita samassa ohjelmassa, niin siirrä piirtämisen välissä kuvioita eri paikkoihin **translate(x,y,z);** komennon avulla. translate(x,y,z); komento siirtää origon uuteen paikkaan. Katsotaan asiaa esimerkin avulla.

```
function setup() {
  createCanvas(800, 800, WEBGL); // Ikkunan koko 800 x 800
}

function draw() {
  background(220); // Taustaväri harmaa
  lights(); // Valot ja varjot käytössä
  rotateX(radians(-mouseY)); // Pyöritä kuviota hiiren avulla
  rotateZ(radians(mouseX)); // Pyöritä kuviota hiiren avulla
  fill(0,255,0); // Aseta vihreä väri
  cylinder(100,200); // Piirrä ellipsi
  translate(0,150,0); // Siirrä origo y-akselia pitkin 150
  fill(255,0,255); // Aseta magenta väri
  sphere(100); // Piirrä pallo, jonka säde on 100
  translate(0,100,0); // Siirrä origo y-akselia pitkin 100
  fill(255,255,0); // Aseta keltainen väri
  cone(100,50); // Piirrä kartio
}
```

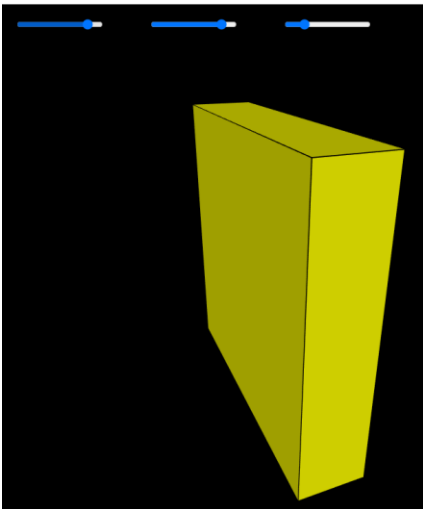


LIUKUPALKIT MUKAAN

```
var a, b, c; // Määritellään muuttujat a, b ja c
function setup() {
  createCanvas(800, 800, WEBGL);
  a = createSlider(0, 500, 100); // Määritellään liukupalkki a
  a.position(20, 20); // Piirrä liukupalkki a kohtaan (20,20)
  b = createSlider(0, 500, 200); // Määritellään liukupalkki b
  b.position(220, 20); // Piirrä liukupalkki b kohtaan (220,20)
  c = createSlider(0, 500, 300); // Määritellään liukupalkki c
  c.position(420, 20); // Piirrä liukupalkki c kohtaan (420,20)
}

function draw() {
  var pituus = a.value(); // Määrittele muuttuja pituus ja sijoita
  // liukupalkin a arvo pituus muuttujalle
  var leveys = b.value(); // Määrittele muuttuja leveys ja sijoita
  // liukupalkin b arvo leveys muuttujalle
  var korkeus = c.value(); // Määrittele muuttuja korkeus ja sijoita
  // liukupalkin c arvo korkeus muuttujalle

  background(0); // Taustaväri musta
  fill(255,255,0); // Täyttöväri keltainen
  lights(); // Luo valot ja varjot
  rotateX(radians(-mouseY)); // Pyöritä x-akselin ympäri
  rotateY(radians(mouseX)); // Pyöritä y-akselin ympäri
  box(pituus, leveys, korkeus); // Piirrä suorakulmainen särmiö
  // muuttujien avulla
}
}
```

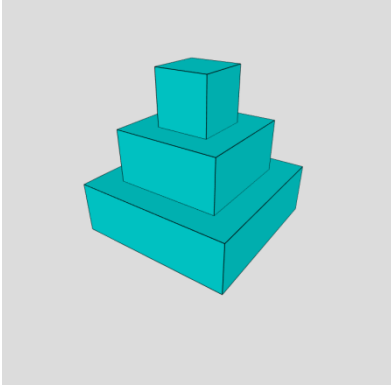


Kun määrittelet ohjelmaan muuttujat (variables), niin voit tallentaa muuttujiin arvoja esimerkiksi liukupalkeista. Tällöin voit muuttaa kuvion kokoa kätevästi liukupalkin avulla.

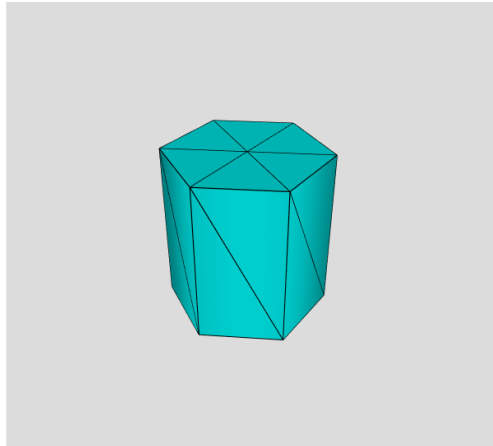
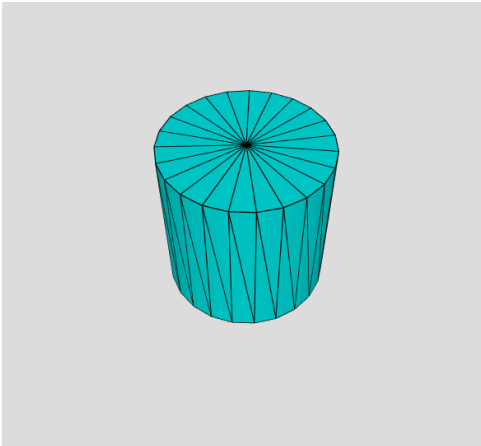
Liukupalkki määritellään `setup()`-lohkossa `createSlider(min,max,oletus);` komennolla ja asetetaan tiettyyn paikkaan `position(x,y);` komennolla. Lukuarvon lukeminen liukupalkista tapahtuu kuitenkin `draw()`-lohkossa `value();` komennolla. Jokaisella eri liukupalkilla on oma muuttujansa, johon liukupalkin arvo tallennetaan.

HARJOITUKSIA

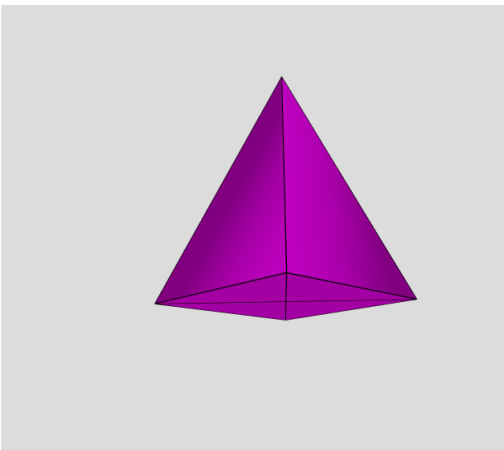
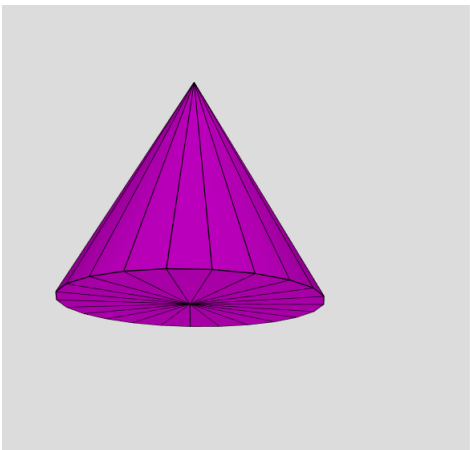
1. Tee ohjelma, joka piirtää kolme erikokoista suorakulmaista särmiötä päällekkäin.



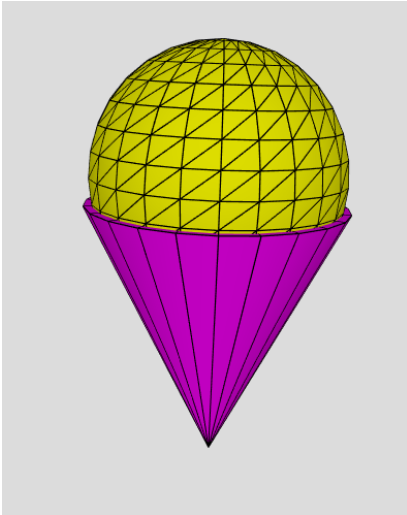
2. Tee ohjelma, joka ympyräpohjaisen lieriön, joka säde 100 ja korkeus 200. Kuinka muutat ympyräpohjaisen lieriön särmiöksi, jossa pohja on kuusikulmio?



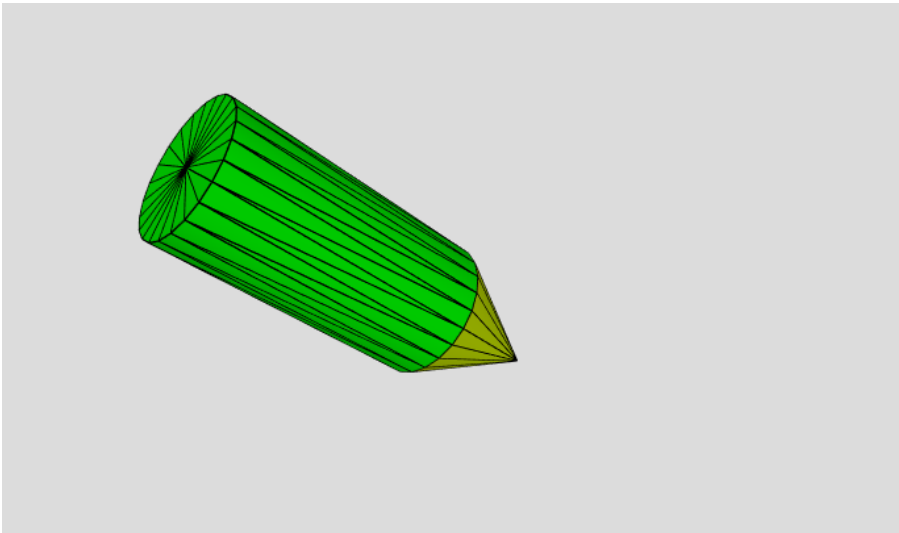
3. Piirrä ympyräpohjainen kartio, jossa pohjan halkaisija ja korkeus ovat molemmat 300. Kuinka muutat pohjan neliöksi, jolloin siitä tulee pyramidi.



4. Tee ohjelma, joka piirtää kartion ja pallon päällekkäin, jolloin syntyy jäätelötötterö.



5. Tee ohjelma, joka piirtää kynän eli yhdistää ympyräpohjaisen lieriön ja kartion.



6. Tee ohjelma, jossa liukupalkkien avulla voidaan säätää kartion sädettä, korkeutta ja tahkojen määrää.

