

TIKZ OPETTAJIEN TARPEISIIN

Ari Heimonen

Oulun normaalikoulu

May 6, 2020

TikZ/PGF-pakettien yleiskuvaus

\LaTeX :ssa voi ladata laajat grafiikkapaketit TikZ ja PGF. Itse asiassa ne eivät ole erillisiä paketteja, vaan kyseessä on saman paketin eri tasot, jotka voi ladata käyttöön myös erillisinä paketteina. Koko grafiikkapaketti sisältää kolme eri tasoa. Peruskäyttäjän tuskin kuitenkaan tarvitseen tietää sen enempää näistä tasoista kuin että TikZ on PGF-grafiikkaohjelmointipaketin edustaohjelma, joka on pyritty tekemään käytöltään helpoksi. Sen verran on syytä huomauttaa, että grafiikkapaketin massiivista manuaalia selatessa tulee vastaan useita eri tapoja tehdä sama asia. Tämä johtuu juuri näistä useista tasoista ja peruskäyttäjälle riittää osata tehdä asia yhdellä tavalla.

\LaTeX :ssa voi ladata laajat grafiikkapaketit TikZ ja PGF. Itse asiassa ne eivät ole erillisiä paketteja, vaan kyseessä on saman paketin eri tasot, jotka voi ladata käyttöön myös erillisinä paketteina. Koko grafiikkapaketti sisältää kolme eri tasoa. Peruskäyttäjän tuskin kuitenkaan tarvitseen tietää sen enempää näistä tasoista kuin että TikZ on PGF-grafiikkaohjelmointipaketin edustaohjelma, joka on pyritty tekemään käytöltään helpoksi. Sen verran on syytä huomauttaa, että grafiikkapaketin massiivista manuaalia selatessa tulee vastaan useita eri tapoja tehdä sama asia. Tämä johtuu juuri näistä useista tasoista ja peruskäyttäjälle riittää osata tehdä asia yhdellä tavalla.

TikZ-nimi tulee saksan kielisestä lauseesta “TikZ ist kein Zeichnenprogman“, joka tahtoo sanoa, että kyseessä ei ole hiirellä piirtämiseen tarkoitettu ohjelma, vaan grafiikkaohjelmointikieli. Ohjelmointikieli on kuitenkin hyvin korkean tason kieli, jossa on paljon valmiita grafiikkaelementtejä ja jopa hyvin arvattavissa olevia toimintoja.

Dokumentin alussa tulee ladata paketit komennolla
`\usepackage{tikz,pgf}`.

TikZ/PGF:n käyttöönotto

Dokumentin alussa tulee ladata paketit komennolla

```
\usepackage{tikz,pgf}.
```

Itse kuvanteko aloitetaan komennolla

```
\begin{tikzpicture}[<optiot>]
```

Dokumentin alussa tulee ladata paketit komennolla

```
\usepackage{tikz,pgf}.
```

Itse kuvanteko aloitetaan komennolla

```
\begin{tikzpicture}[<optiot>]
```

ja lopetaan komennolla

```
\end{tikzpicture}.
```

Dokumentin alussa tulee ladata paketit komennolla

```
\usepackage{tikz,pgf}.
```

Itse kuvanteko aloitetaan komennolla

```
\begin{tikzpicture}[<optiot>]
```

ja lopetaan komennolla

```
\end{tikzpicture}.
```

Hyvin käytännöllinen piirre TikZ:issä on se, että optioita voi antaa monessa eri vaiheessa riippuen siitä haluaako niiden olevan voimassa koko kuvassa, jonkin komennon vaikutusalueessa tai yksittäisessä piirtoelementissä. Ylläoleessa paikassa annetaan siis optiot, jotka ovat voimassa koko kuvassa.

Dokumentin alussa tulee ladata paketit komennolla

```
\usepackage{tikz,pgf}.
```

Itse kuvanteko aloitetaan komennolla

```
\begin{tikzpicture}[<optiot>]
```

ja lopetaan komennolla

```
\end{tikzpicture}.
```

Hyvin käytännöllinen piirre TikZ:issä on se, että optioita voi antaa monessa eri vaiheessa riippuen siitä haluaako niiden olevan voimassa koko kuvassa, jonkin komennon vaikutusalueessa tai yksittäisessä piirtoelementissä. Ylläoleessa paikassa annetaan siis optiot, jotka ovat voimassa koko kuvassa.

Tällaisia optioita ovat esimerkiksi

- Koordinaattien yksiköt, esimerkiksi $x=0.5\text{cm}$, $y=2\text{cm}$

Dokumentin alussa tulee ladata paketit komennolla

```
\usepackage{tikz,pgf}.
```

Itse kuvanteko aloitetaan komennolla

```
\begin{tikzpicture}[<optiot>]
```

ja lopetaan komennolla

```
\end{tikzpicture}.
```

Hyvin käytännöllinen piirre TikZ:issä on se, että optioita voi antaa monessa eri vaiheessa riippuen siitä haluaako niiden olevan voimassa koko kuvassa, jonkin komennon vaikutusalueessa tai yksittäisessä piirtoelementissä. Ylläoleessa paikassa annetaan siis optiot, jotka ovat voimassa koko kuvassa.

Tällaisia optioita ovat esimerkiksi

- Koordinaattien yksiköt, esimerkiksi $x=0.5\text{cm}$, $y=2\text{cm}$
- Skaala $scale=<luku>$, jolla voi kerran piirrettyä kuvaa suurentaa tai pienentää.

Esimerkiksi Pekka on piirtänyt sinifunktion kuvaajan edellämainittuja yksikköjä käyttäen ja huomaa sitten, että kuva ei sovi siihen, mihin se on tarkoitettu. Tällöin Pekka lisää esimerkiksi option `scale= 0.7`

Esimerkiksi Pekka on piirtänyt sinifunktion kuvaajan edellämainittuja yksikköjä käyttäen ja huomaa sitten, että kuva ei sovi siihen, mihin se on tarkoitettu. Tällöin Pekka lisää esimerkiksi option `scale= 0.7` eli kokonaisuudessaan

```
\begin{tikzpicture}[x=0.5cm,y=2cm,scale=0.7].
```

Esimerkiksi Pekka on piirtänyt sinifunktion kuvaajan edellämainittuja yksikköjä käyttäen ja huomaa sitten, että kuva ei sovi siihen, mihin se on tarkoitettu. Tällöin Pekka lisää esimerkiksi option `scale=0.7` eli kokonaisuudessaan

```
\begin{tikzpicture}[x=0.5cm,y=2cm,scale=0.7].
```

Peruskomento TikZ:issa on

```
\draw[<optiot>] <piirtotoiminnot>;
```

Esimerkiksi Pekka on piirtänyt sinifunktion kuvaajan edellämainittuja yksikköjä käyttäen ja huomaa sitten, että kuva ei sovi siihen, mihin se on tarkoitettu. Tällöin Pekka lisää esimerkiksi option `scale=0.7` eli kokonaisuudessaan

```
\begin{tikzpicture}[x=0.5cm,y=2cm,scale=0.7].
```

Peruskomento TikZ:issa on

```
\draw[<optiot>] <piirtotoiminnot>;
```

Tässä puolipiste lopussa on huomionarvoinen asia, koska sen unohtaminen aiheuttaa yleensä **fatal error**-ilmoituksen eikä L^AT_EX tuota yhtään sivua.

Esimerkiksi Pekka on piirtänyt sinifunktion kuvaajan edellämäinittuja yksikköjä käyttäen ja huomaa sitten, että kuva ei sovi siihen, mihin se on tarkoitettu. Tällöin Pekka lisää esimerkiksi option `scale=0.7` eli kokonaisuudessaan

```
\begin{tikzpicture}[x=0.5cm,y=2cm,scale=0.7].
```

Peruskomento TikZ:issa on

```
\draw[<optiot>] <piirtotoiminnot>;
```

Tässä puolipiste lopussa on huomionarvoinen asia, koska sen unohtaminen aiheuttaa yleensä **fatal error**-ilmoituksen eikä L^AT_EX tuota yhtään sivua.

Piirtotoiminto alkaa tyypillisesti pisteen, nk. *current point*, koordinaateilla jotka annetaan normaalisti, esimerkiksi **(2,3)**, sen jälkeen se, mitä halutaan piirtää ja siihen tarvittava lisäinformaatio.

Keskeisiä piirtoelementtejä ovat

Keskeisiä piirtoelementtejä ovat

- $(x_1, y_1) - - (x_2, y_2)$ piirtää suoran viivan pisteestä (x_1, y_1) pisteeseen (x_2, y_2) .

Keskeisiä piirtoelementtejä ovat

- $(x_1, y_1) - - (x_2, y_2)$ piirtää suoran viivan pisteestä (x_1, y_1) pisteeseen (x_2, y_2) .
- $(x_1, y_1) - | (x_2, y_2)$ yhdistää pisteet viivalla, joka on ensin vaakasuora ja sitten pystysuora.

Keskeisiä piirtoelementtejä ovat

- $(x_1, y_1) - - (x_2, y_2)$ piirtää suoran viivan pisteestä (x_1, y_1) pisteeseen (x_2, y_2) .
- $(x_1, y_1) - | (x_2, y_2)$ yhdistää pisteet viivalla, joka on ensin vaakasuora ja sitten pystysuora.
- (x_1, y_1) **rectangle** (x_2, y_2) piirtää suorakaiteen, jonka vastakkaiset kulmapisteet ovat (x_1, y_1) ja (x_2, y_2) .

Keskeisiä piirtoelementtejä ovat

- $(x_1, y_1) - - (x_2, y_2)$ piirtää suoran viivan pisteestä (x_1, y_1) pisteeseen (x_2, y_2) .
- $(x_1, y_1) - | (x_2, y_2)$ yhdistää pisteet viivalla, joka on ensin vaakasuora ja sitten pystysuora.
- (x_1, y_1) `rectangle` (x_2, y_2) piirtää suorakaiteen, jonka vastakkaiset kulmapisteet ovat (x_1, y_1) ja (x_2, y_2) .
- (x_1, y_1) `circle` (`<säde>`) piirtää ympyrän keskipisteenä (x_1, y_1) .

Keskeisiä piirtoelementtejä ovat

- $(x_1, y_1) - - (x_2, y_2)$ piirtää suoran viivan pisteestä (x_1, y_1) pisteeseen (x_2, y_2) .
- $(x_1, y_1) - | (x_2, y_2)$ yhdistää pisteet viivalla, joka on ensin vaakasuora ja sitten pystysuora.
- (x_1, y_1) **rectangle** (x_2, y_2) piirtää suorakaiteen, jonka vastakkaiset kulmapisteet ovat (x_1, y_1) ja (x_2, y_2) .
- (x_1, y_1) **circle** \langle säde \rangle piirtää ympyrän keskipisteenä (x_1, y_1) .
- (x_1, y_1) **arc** \langle alkukulma \rangle : \langle loppukulma \rangle : \langle säde \rangle piirtää ympyränkaaren alkaen pisteestä (x_1, y_1) (ks. tarkemmin esimerkki myöhemmin).

Keskeisiä piirtoelementtejä ovat

- $(x_1, y_1) - - (x_2, y_2)$ piirtää suoran viivan pisteestä (x_1, y_1) pisteeseen (x_2, y_2) .
- $(x_1, y_1) - | (x_2, y_2)$ yhdistää pisteet viivalla, joka on ensin vaakasuora ja sitten pystysuora.
- (x_1, y_1) **rectangle** (x_2, y_2) piirtää suorakaiteen, jonka vastakkaiset kulmapisteet ovat (x_1, y_1) ja (x_2, y_2) .
- (x_1, y_1) **circle** \langle säde \rangle piirtää ympyrän keskipisteenä (x_1, y_1) .
- (x_1, y_1) **arc** \langle alkukulma \rangle : \langle loppukulma \rangle : \langle säde \rangle piirtää ympyränkaaren alkaen pisteestä (x_1, y_1) (ks. tarkemmin esimerkki myöhemmin).
- (x_1, y_1) **node** $\{ \langle$ teksti $\rangle \}$ sijoittaa \langle teksti \rangle :n pisteeseen (x_1, y_1) .

Seuraaviin esimerkkeihin on piirretty havainnollisuuden vuoksi koordinaattiakselit ja ruudukko. Ne opitaan piirtämään hieman myöhemmin.

TikZ-ympäristön sisällä annetut komennot

Seuraaviin esimerkkeihin on piirretty havainnollisuuden vuoksi koordinaattiakselit ja ruudukko. Ne opitaan piirtämään hieman myöhemmin.

TikZ-ympäristön sisällä annetut komennot

```
\draw (1,1) -- (4,3);
```

```
\draw (1,1) --| (4,3);
```

piirtävät kolmion:

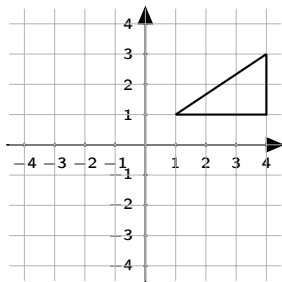
Seuraaviin esimerkkeihin on piirretty havainnollisuuden vuoksi koordinaattiakselit ja ruudukko. Ne opitaan piirtämään hieman myöhemmin.

TikZ-ympäristön sisällä annetut komennot

```
\draw (1,1) -- (4,3);
```

```
\draw (1,1) --| (4,3);
```

piirtävät kolmion:



Seuraaviin esimerkkeihin on piirretty havainnollisuuden vuoksi koordinaattiakselit ja ruudukko. Ne opitaan piirtämään hieman myöhemmin.

TikZ-ympäristön sisällä annetut komennot

```
\draw (1,1) -- (4,3);
```

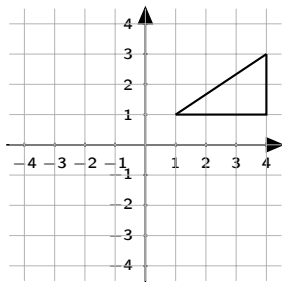
```
\draw (1,1) -- | (4,3);
```

piirtävät kolmion:

```
\draw (-3,-2) rectangle (-1,1);
```

```
\draw (2,2) circle (2cm);
```

piirtävät suorakulmion ja ympyrän:



Esimerkkejä

Seuraaviin esimerkkeihin on piirretty havainnollisuuden vuoksi koordinaattiakselit ja ruudukko. Ne opitaan piirtämään hieman myöhemmin.

TikZ-ympäristön sisällä annetut komennot

```
\draw (1,1) -- (4,3);
```

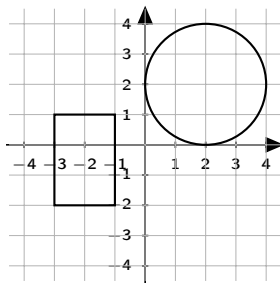
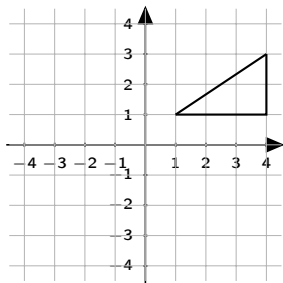
```
\draw (1,1) -- | (4,3);
```

piirtävät kolmion:

```
\draw (-3,-2) rectangle (-1,1);
```

```
\draw (2,2) circle (2cm);
```

piirtävät suorakulmion ja ympyrän:



Kaaren piirtoelementin ymmärtämiseksi seuraaviin kuviin on piirretty pisteviivalla se ympyrä tai ellipsi, jonka kaaren elementti piirtää.

Kaaren piirtoelementin ymmärtämiseksi seuraaviin kuviin on piirretty pisteviivalla se ympyrä tai ellipsi, jonka kaaren elementti piirtää.

Komento

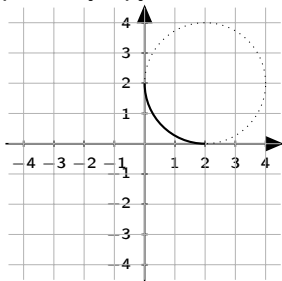
```
\draw  
(0,2) arc (180:270:2cm);
```

Kaaren piirtoelementin ymmärtämiseksi seuraaviin kuviin on piirretty pisteiviivalla se ympyrä tai ellipsi, jonka kaaren elementti piirtää.

Komento

```
\draw  
(0,2) arc (180:270:2cm);
```

piirtää ympyrän kaaren

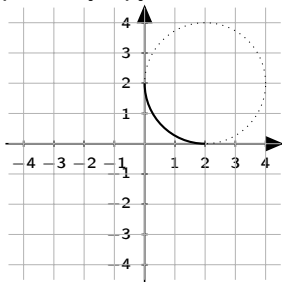


Kaaren piirtoelementin ymmärtämiseksi seuraaviin kuviin on piirretty pisteviivalla se ympyrä tai ellipsi, jonka kaaren elementti piirtää.

Komento

```
\draw  
(0,2) arc (180:270:2cm);
```

piirtää ympyrän kaaren



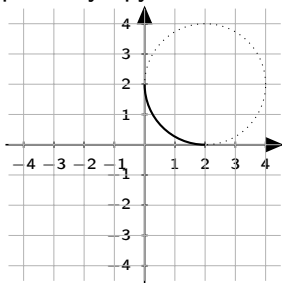
Komento

```
\draw  
(2,2.5) arc (90:180:2cm and 0.5cm);
```

Kaaren piirtoelementin ymmärtämiseksi seuraaviin kuviin on piirretty pisteviivalla se ympyrä tai ellipsi, jonka kaaren elementti piirtää.

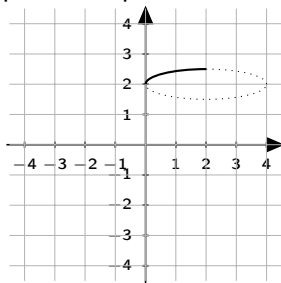
Komento

`\draw`
`(0,2) arc (180:270:2cm);`
piirtää ympyrän kaaren



Komento

`\draw`
`(2,2.5) arc (90:180:2cm and 0.5cm);`
piirtää ellipsin kaaren



Samoin kuin kaaren suhteen, **circle**-elementti piirtää ellipsin, jos annetaan kaksi sädettä (ts. akselien puolikkaat). Huomaa sana **and**. Avaruuskuvien tekemisessä tarvitaan usein ellipsin kaaria, esimerkiksi sylinteri voidaan piirtää seuraavasti:

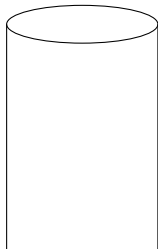
Samoin kuin kaaren suhteen, `circle`-elementti piirtää ellipsin, jos annetaan kaksi sädettä (ts. akselien puolikkaat). Huomaa sana `and`. Avaruuskuvien tekemisessä tarvitaan usein ellipsin kaaria, esimerkiksi sylinteri voidaan piirtää seuraavasti:

```
\draw (0,0) – (0,3);  
\draw(2,0) –(2,3);
```



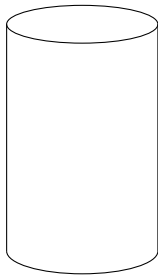
Samoin kuin kaaren suhteen, `circle`-elementti piirtää ellipsin, jos annetaan kaksi sädettä (ts. akselien puolikkaat). Huomaa sana `and`. Avaruuskuvien tekemisessä tarvitaan usein ellipsin kaaria, esimerkiksi sylinteri voidaan piirtää seuraavasti:

```
\draw (0,0) – (0,3);  
\draw(2,0) –(2,3);  
\draw (1,3) circle (1cm and 0.25cm);
```



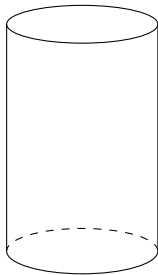
Samoin kuin kaaren suhteen, `circle`-elementti piirtää ellipsin, jos annetaan kaksi sädettä (ts. akselien puolikkaat). Huomaa sana `and`. Avaruuskuvien tekemisessä tarvitaan usein ellipsin kaaria, esimerkiksi sylinteri voidaan piirtää seuraavasti:

```
\draw (0,0) – (0,3);  
\draw(2,0) –(2,3);  
\draw (1,3) circle (1cm and 0.25cm);  
\draw (0,0) arc (180:360:1cm and  
0.3cm);
```



Samoin kuin kaaren suhteen, `circle`-elementti piirtää ellipsin, jos annetaan kaksi sädettä (ts. akselien puolikkaat). Huomaa sana `and`. Avaruuskuvien tekemisessä tarvitaan usein ellipsin kaaria, esimerkiksi sylinteri voidaan piirtää seuraavasti:

```
\draw (0,0) – (0,3);  
\draw(2,0) –(2,3);  
\draw (1,3) circle (1cm and 0.25cm);  
\draw (0,0) arc (180:360:1cm and  
0.3cm);  
\draw[dashed] (2,0) arc (0:180:1cm  
and 0.3cm);
```



Yhdistämällä useita viivan piirtoelementtejä niin, että seuraava alkaa siitä pisteestä, mihin edellinen päättyy, saadaan aikaan **polku** (path). Lisäksi jokaiseen pisteeseen voidaan liittää **node**.

Esimerkiksi komentojono

Yhdistämällä useita viivan piirtoelementtejä niin, että seuraava alkaa siitä pisteestä, mihin edellinen päättyy, saadaan aikaan **polku** (path). Lisäksi jokaiseen pisteeseen voidaan liittää **node**.

Esimerkiksi komentojono

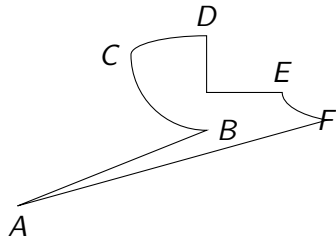
```
\draw (-3,-2)
node[below] { $A$ }- -
(2,0) node[right] { $B$ }
arc (-90:-180:2cm)
node [left] { $C$ }
arc(-180:-270: 2cm and 0.5cm)
node[above] { $D$ } |-
(4,1) node[right] { $E$ } - -
arc (180:246:2cm and 0.8cm)
node { $F$ } - - cycle;
```

Yhdistämällä useita viivan piirtoelementtejä niin, että seuraava alkaa siitä pisteestä, mihin edellinen päättyy, saadaan aikaan **polku** (path). Lisäksi jokaiseen pisteeseen voidaan liittää **node**.

Esimerkiksi komentojono

```
\draw (-3,-2)
node[below] {$A$} - -
(2,0) node[right] {$B$}
arc (-90:-180:2cm)
node [left] {$C$}
arc(-180:-270: 2cm and 0.5cm)
node[above] {$D$} |-
(4,1) node[right] {$E$} - -
arc (180:246:2cm and 0.8cm)
node {$F$} - - cycle;
```

piirtää seuraavan kuvan:



Kuten ehkä huomasitte, Arille oli pujahtanut virhe edellisessä kuvassa.

Kuten ehkä huomasitte, Arille oli pujahtanut virhe edellisessä kuvassa. Kirjain F oli kirjoitettu pisteen F päälle.

Kuten ehkä huomasitte, Arille oli pujahtanut virhe edellisessä kuvassa. Kirjain F oli kirjoitettu pisteen F päälle. Ellei toisin määrätä, TikZ sijoittaa tekstin keskipisteen pisteeseen `current point`.

Kuten ehkä huomasitte, Arille oli pujahtanut virhe edellisessä kuvassa. Kirjain F oli kirjoitettu pisteen F päälle. Ellei toisin määrätä, TikZ sijoittaa tekstin keskipisteen pisteeseen `current point`. Toisin määrittäminen tapahtuu optioiden avulla. Nämä optiot ovat **left**, **right**, **below** ja **above**.

Kuten ehkä huomasitte, Arille oli pujahtanut virhe edellisessä kuvassa. Kirjain F oli kirjoitettu pisteen F päälle. Ellei toisin määrätä, TikZ sijoittaa tekstin keskipisteen pisteeseen `current point`. Toisin määrittäminen tapahtuu optioiden avulla. Nämä optiot ovat **left**, **right**, **below** ja **above**.

Piirtoelementtien optioita:

Kuten ehkä huomasitte, Arille oli pujahtanut virhe edellisessä kuvassa. Kirjain F oli kirjoitettu pisteen F päälle. Ellei toisin määrätä, TikZ sijoittaa tekstin keskipisteen pisteeseen `current point`. Toisin määrittäminen tapahtuu optioiden avulla. Nämä optiot ovat **left**, **right**, **below** ja **above**.

Piirtoelementtien optioita:

- Väri määrätään optiolla `color=<väri>`. TikZ tuntee ainakin tavallisimmat englanninkieliset värien nimet ja värejä voi myös sekoittaa. Esimerkiksi `red!80!black` tuottaa värin, jossa 80% punaista ja loput mustaa.

Kuten ehkä huomasitte, Arille oli pujahtanut virhe edellisessä kuvassa. Kirjain F oli kirjoitettu pisteen F päälle. Ellei toisin määrätä, TikZ sijoittaa tekstin keskipisteen pisteeseen `current point`. Toisin määrittäminen tapahtuu optioiden avulla. Nämä optiot ovat `left`, `right`, `below` ja `above`.

Piirtoelementtien optioita:

- Väri määrätään optiolla `color=<väri>`. TikZ tuntee ainakin tavallisimmat englanninkieliset värien nimet ja värejä voi myös sekoittaa. Esimerkiksi `red!80!black` tuottaa värin, jossa 80% punaista ja loput mustaa.
- Viivan paksuus voidaan asettaa optioiden `very thin`, `thin`, `semithick`, `thick` ja `very thick` avulla.

Kuten ehkä huomasitte, Arille oli pujahtanut virhe edellisessä kuvassa. Kirjain F oli kirjoitettu pisteen F päälle. Ellei toisin määrätä, TikZ sijoittaa tekstin keskipisteen pisteeseen `current point`. Toisin määrääminen tapahtuu optioiden avulla. Nämä optiot ovat `left`, `right`, `below` ja `above`.

Piirtoelementtien optioita:

- Väri määrätään optiolla `color=<väri>`. TikZ tuntee ainakin tavallisimmat englanninkieliset värien nimet ja värejä voi myös sekoittaa. Esimerkiksi `red!80!black` tuottaa värin, jossa 80% punaista ja loput mustaa.
- Viivan paksuus voidaan asettaa optioiden `very thin`, `thin`, `semithick`, `thick` ja `very thick` avulla.
- Optio `dotted` tekee viivasta pisteviivan.

Kuten ehkä huomasitte, Arille oli pujahtanut virhe edellisessä kuvassa. Kirjain F oli kirjoitettu pisteen F päälle. Ellei toisin määrätä, TikZ sijoittaa tekstin keskipisteen pisteeseen `current point`. Toisin määrääminen tapahtuu optioiden avulla. Nämä optiot ovat `left`, `right`, `below` ja `above`.

Piirtoelementtien optioita:

- Väri määrätään optiolla `color=<väri>`. TikZ tuntee ainakin tavallisimmat englanninkieliset värien nimet ja värejä voi myös sekoittaa. Esimerkiksi `red!80!black` tuottaa värin, jossa 80% punaista ja loput mustaa.
- Viivan paksuus voidaan asettaa optioiden `very thin`, `thin`, `semithick`, `thick` ja `very thick` avulla.
- Optio `dotted` tekee viivasta pisteviivan.
- Optio `dashed` tekee viivasta katkoviivan.

Tekstin sijoittaminen kuviin

Toiminnon `node` avulla voidaan sijoittaa haluttu teksti niin, että nykyinen piste on tekstin keskipisteenä.

Tekstin sijoittaminen kuviin

Toiminnon **node** avulla voidaan sijoittaa haluttu teksti niin, että nykyinen piste on tekstin keskipisteenä. Usein kuitenkin halutaan sijoittaa symboleja, mittoja yms. janan keskipisteen kohdalle. Tämä onnistuu sijoittamalla node merkin - - jälkeen.

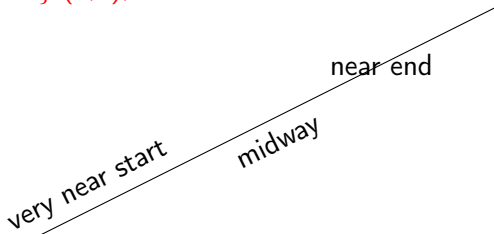
Tekstin sijoittaminen kuviin

Toiminnon `node` avulla voidaan sijoittaa haluttu teksti niin, että nykyinen piste on tekstin keskipisteenä. Usein kuitenkin halutaan sijoittaa symboleja, mittoja yms. janan keskipisteen kohdalle. Tämä onnistuu sijoittamalla `node` merkin - - jälkeen. Lisäksi voidaan teksti kallistaa (optio `sloped`) samaan kaltevuuteen kuin jana.

Tekstin sijoittaminen kuviin

Toiminnon `node` avulla voidaan sijoittaa haluttu teksti niin, että nykyinen piste on tekstin keskipisteenä. Usein kuitenkin halutaan sijoittaa symboleja, mittoja yms. janan keskipisteen kohdalle. Tämä onnistuu sijoittamalla `node` merkin - - jälkeen. Lisäksi voidaan teksti kallistaa (optio `sloped`) samaan kaltevuuteen kuin jana. Seuraavassa esimerkissä on esitetty toiminnon mahdollisia optioita:

```
\draw (0,0) - - node[sloped, very near start,above]{ very near start } node[sloped, midway,below]{ midway } node[near end]{ near end } (6,3);
```



Mittojen sijoittaminen kuviin

Jotta mitat saisi hyvin kuviin, kannattaa ladata myös siunitx-paketti lisäämällä alkuun komento

```
\usepackage[detect-all,output-decimal-marker={,}]{siunitx}
```

Mittojen sijoittaminen kuviin

Jotta mitat saisi hyvin kuviin, kannattaa ladata myös siunitx-paketti lisäämällä alkuun komento

```
\usepackage[detect-all,output-decimal-marker={,}]{siunitx}
```

Koulumatematiikassa on käytössä desimaalipilkku ja sen käytössä ilman ko. pakettia tulee sellainen ongelma, että \LaTeX lisää pilkun yhteyteen ylimääräisen välin. Lisäksi koodiin lisätty optio muuttaa automaattisesti desimaalipisteen pilkuksi, joten arvoja voi laskea toisella ohjelmalla ja lisätä ne suoraan copy-paste-toiminnolla koodiin.

Mittojen sijoittaminen kuviin

Jotta mitat saisi hyvin kuviin, kannattaa ladata myös siunitx-paketti lisäämällä alkuun komento

```
\usepackage[detect-all,output-decimal-marker={,}]{siunitx}
```

Koulumatematiikassa on käytössä desimaalipilkku ja sen käytössä ilman ko. pakettia tulee sellainen ongelma, että \LaTeX lisää pilkun yhteyteen ylimääräisen välin. Lisäksi koodiin lisätty optio muuttaa automaattisesti desimaalipisteen pilkuksi, joten arvoja voi laskea toisella ohjelmalla ja lisätä ne suoraan copy-paste-toiminnolla koodiin.

Paketin avulla myös lukujen ja yksikköjen välistys tulee oikean näköiseksi. Edelleen paketissa on komennot monille yksiköille, kuten kulma-aste ja Celsius-aste, joita ei perusversiosta löydy. Lisäksi paketti lisää suuriin lukuihin automaattisesti pienen välin tuhanerottimeksi.

Mittojen sijoittaminen kuviin

Jotta mitat saisi hyvin kuviin, kannattaa ladata myös siunitx-paketti lisäämällä alkuun komento

```
\usepackage[detect-all,output-decimal-marker={,}]{siunitx}
```

Koulumatematiikassa on käytössä desimaalipilkku ja sen käytössä ilman ko. pakettia tulee sellainen ongelma, että \LaTeX lisää pilkun yhteyteen ylimääräisen välin. Lisäksi koodiin lisätty optio muuttaa automaattisesti desimaalipisteen pilkuksi, joten arvoja voi laskea toisella ohjelmalla ja lisätä ne suoraan copy-paste-toiminnolla koodiin.

Paketin avulla myös lukujen ja yksikköjen välistys tulee oikean näköiseksi. Edelleen paketissa on komennot monille yksiköille, kuten kulma-aste ja Celsius-aste, joita ei perusversiosta löydy. Lisäksi paketti lisää suuriin lukuihin automaattisesti pienen välin tuhanerottimeksi.

Paketin muihin mahdollisuuksiin voi tutustua manuaalin avulla. Linkki löytyy pedanet-sivuilta.

Mittojen sijoittaminen kuviin

Yksiköttämät luvut kirjoitetaan

`\num{<luku>}`.

Mittojen sijoittaminen kuviin

Yksiköttämät luvut kirjoitetaan

`\num{<luku>}`.

Esimerkiksi `\num{2.34}` antaa tulostuksen 2,34 ja `\num{234567678}` tulostuksen 234 567 678.

Mittojen sijoittaminen kuviin

Yksiköttämät luvut kirjoitetaan

`\num{<luku> }`.

Esimerkiksi `\num{2.34}` antaa tulostuksen 2,34 ja `\num{234567678}` tulostuksen 234 567 678.

Yksikölliset luvut kirjoitetaan

`\SI{<luku> }{<yksikkö> }`.

Mittojen sijoittaminen kuviin

Yksiköttämät luvut kirjoitetaan

`\num{<luku>}`.

Esimerkiksi `\num{2.34}` antaa tulostuksen 2,34 ja `\num{234567678}` tulostuksen 234 567 678.

Yksikölliset luvut kirjoitetaan

`\SI{<luku>}{<yksikkö>}`.

Esimerkiksi `\SI{2.34}{cm}` antaa tulostuksen 2,34 cm ja `\SI{56,8}{cm^2}` tulostuksen 56,8 cm².

Mittojen sijoittaminen kuviin

Yksiköttämät luvut kirjoitetaan

`\num{<luku>}`.

Esimerkiksi `\num{2.34}` antaa tulostuksen 2,34 ja `\num{234567678}` tulostuksen 234 567 678.

Yksikölliset luvut kirjoitetaan

`\SI{<luku>}{<yksikkö>}`.

Esimerkiksi `\SI{2.34}{cm}` antaa tulostuksen 2,34 cm ja `\SI{56,8}{cm^2}` tulostuksen 56,8 cm².

Kulma kirjoitetaan

`\ang{<arvo>}`

Mittojen sijoittaminen kuviin

Yksiköttämät luvut kirjoitetaan

`\num{<luku>}`.

Esimerkiksi `\num{2.34}` antaa tulostuksen 2,34 ja `\num{234567678}` tulostuksen 234 567 678.

Yksikölliset luvut kirjoitetaan

`\SI{<luku>}{<yksikkö>}`.

Esimerkiksi `\SI{2.34}{cm}` antaa tulostuksen 2,34 cm ja `\SI{56,8}{cm^2}` tulostuksen 56,8 cm².

Kulma kirjoitetaan

`\ang{<arvo>}`

Esimerkiksi `\ang{48,6}` tuottaa 48,6°.

Mittojen sijoittaminen kuviin

Yksiköttämät luvut kirjoitetaan

`\num{<luku>}`.

Esimerkiksi `\num{2.34}` antaa tulostuksen 2,34 ja `\num{234567678}` tulostuksen 234 567 678.

Yksikölliset luvut kirjoitetaan

`\SI{<luku>}{<yksikkö>}`.

Esimerkiksi `\SI{2.34}{cm}` antaa tulostuksen 2,34 cm ja `\SI{56,8}{cm^2}` tulostuksen 56,8 cm².

Kulma kirjoitetaan

`\ang{<arvo>}`

Esimerkiksi `\ang{48,6}` tuottaa 48,6°.

Celsius-asteet kirjoitetaan

`\SI{<arvo>}{\celsius}`

Mittojen sijoittaminen kuvaan

Yksiköttämät luvut kirjoitetaan

`\num{<luku>}`.

Esimerkiksi `\num{2.34}` antaa tulostuksen 2,34 ja `\num{234567678}` tulostuksen 234 567 678.

Yksikölliset luvut kirjoitetaan

`\SI{<luku>}{<yksikkö>}`.

Esimerkiksi `\SI{2.34}{cm}` antaa tulostuksen 2,34 cm ja `\SI{56,8}{cm^2}` tulostuksen 56,8 cm².

Kulma kirjoitetaan

`\ang{<arvo>}`

Esimerkiksi `\ang{48,6}` tuottaa 48,6°.

Celsius-asteet kirjoitetaan

`\SI{<arvo>}{\celsius}`

Esimerksi `\SI{-237,15}{\celsius}` tuottaa -237,15 °C

Täyttäminen värillä ja varjostaminen

Suljetun polun sisäpuolen voi täyttää haluamallaan värillä komennolla `\fill[color=<väri>] <polku>;`.

Täyttäminen värillä ja varjostaminen

Suljetun polun sisäpuolen voi täyttää haluamallaan värillä komennolla `\fill[color=<väri>] <polku>;`. Huomaa, että piirtojärjestys määrää, mikä elementti tulee päällimmäiseksi. Jos värillä täytetty alue piirretään viimeiseksi, se peittää kaikki alleen jäävät piirtoelementit.

Täyttäminen värillä ja varjostaminen

Suljetun polun sisäpuolen voi täyttää haluamallaan värillä komennolla `\fill[color=<väri>] <polku>;`. Huomaa, että piirtojärjestys määrää, mikä elementti tulee päällimmäiseksi. Jos värillä täytetty alue piirretään viimeiseksi, se peittää kaikki alleen jäävät piirtoelementit. Elementeille voi myös asettaa läpinäkyvyyttä optiolla `opacity=<luku>`, mutta tällöin myös väri vaalenee.

Täyttäminen värillä ja varjostaminen

Suljetun polun sisäpuolen voi täyttää haluamallaan värillä komennolla `\fill[color=<väri>] <polku>;`. Huomaa, että piirtojärjestys määrää, mikä elementti tulee päällimmäiseksi. Jos värillä täytetty alue piirretään viimeiseksi, se peittää kaikki alleen jäävät piirtoelementit. Elementeille voi myös asettaa läpinäkyvyyttä optiolla `opacity=<luku>`, mutta tällöin myös väri vaalenee. Avaruuskuviiin saa kolmiulotteisuutta varjoksella, joka tapahtuu komennolla `\shade<optiot> <polku>;`.

Täyttäminen värillä ja varjostaminen

Suljetun polun sisäpuolen voi täyttää haluamallaan värillä komennolla `\fill[color=<väri>] <polku>;`. Huomaa, että piirtojärjestys määrää, mikä elementti tulee päällimmäiseksi. Jos värillä täytetty alue piirretään viimeiseksi, se peittää kaikki alleen jäävät piirtoelementit. Elementeille voi myös asettaa läpinäkyvyyttä optiolla `opacity=<luku>`, mutta tällöin myös väri vaalenee. Avaruuskuviin saa kolmiulotteisuutta varjoksella, joka tapahtuu komennolla `\shade[<optiot>] <polku>;`. Optioina voi antaa varjostusvärin (tai -värit) ja varjostustavan, joka voi olla `ball`, `axis` tai `radial`.

Täyttäminen värillä ja varjostaminen

Suljetun polun sisäpuolen voi täyttää haluamallaan värillä komennolla `\fill[color=<väri>] <polku>`; . Huomaa, että piirtojärjestys määrää, mikä elementti tulee päällimmäiseksi. Jos värillä täytetty alue piirretään viimeiseksi, se peittää kaikki alleen jäävät piirtoelementit. Elementeille voi myös asettaa läpinäkyvyyttä optiolla `opacity=<luku>`, mutta tällöin myös väri vaalenee.

Avaruuskuviiin saa kolmiulotteisuutta varjoksella, joka tapahtuu komennolla `\shade[<optiot>] <polku>`; . Optioina voi antaa varjostusvärin (tai -värit) ja varjostustavan, joka voi olla `ball`, `axis` tai `radial`.

Varjostaminen `axis`-optiolla tapahtuu antamalla varjostusvärit optioilla joko `top/middle/bottom color=< väri>` tai `left/middle/right color=< väri>`.

Täyttäminen värillä ja varjostaminen

Suljetun polun sisäpuolen voi täyttää haluamallaan värillä komennolla `\fill[color=<väri>] <polku>`; . Huomaa, että piirtojärjestys määrää, mikä elementti tulee päällimmäiseksi. Jos värillä täytetty alue piirretään viimeiseksi, se peittää kaikki alleen jäävät piirtoelementit. Elementeille voi myös asettaa läpinäkyvyyttä optiolla `opacity=<luku>`, mutta tällöin myös väri vaalenee.

Avaruuskuviin saa kolmiulotteisuutta varjoksella, joka tapahtuu komennolla `\shade[<optiot>] <polku>`; . Optioina voi antaa varjostusvärin (tai -värit) ja varjostustavan, joka voi olla `ball`, `axis` tai `radial`.

Varjostaminen `axis`-optiolla tapahtuu antamalla varjostusvärit optioilla joko `top/middle/bottom color=<väri>` tai `left/middle/right color=<väri>`. Näistä yksi tai kaksi voidaan jättää pois, jolloin oletusarvo on valkoinen.

Täyttäminen värillä ja varjostaminen

Suljetun polun sisäpuolen voi täyttää haluamallaan värillä komennolla `\fill[color=<väri>] <polku>`; . Huomaa, että piirtojärjestys määrää, mikä elementti tulee päällimmäiseksi. Jos värillä täytetty alue piirretään viimeiseksi, se peittää kaikki alleen jäävät piirtoelementit. Elementeille voi myös asettaa läpinäkyvyyttä optiolla `opacity=<luku>`, mutta tällöin myös väri vaalenee.

Avaruuskuviiin saa kolmiulotteisuutta varjoksella, joka tapahtuu komennolla `\shade[<optiot>] <polku>`; . Optioina voi antaa varjostusvärin (tai -värit) ja varjostustavan, joka voi olla `ball`, `axis` tai `radial`.

Varjostaminen `axis`-optiolla tapahtuu antamalla varjostusvärit optioilla joko `top/middle/bottom color=< väri>` tai `left/middle/right color=< väri>`. Näistä yksi tai kaksi voidaan jättää pois, jolloin oletusarvo on valkoinen. Vastaavasti `radial`-varjostaminen tapahtuu antamalla `inner/outer color=< väri>`.

Täyttäminen värillä ja varjostaminen

Suljetun polun sisäpuolen voi täyttää haluamallaan värillä komennolla `\fill[color=<väri>] <polku>;`. Huomaa, että piirtojärjestys määrää, mikä elementti tulee päällimmäiseksi. Jos värillä täytetty alue piirretään viimeiseksi, se peittää kaikki alleen jäävät piirtoelementit. Elementeille voi myös asettaa läpinäkyvyyttä optiolla `opacity=<luku>`, mutta tällöin myös väri vaalenee.

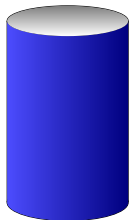
Avaruuskuvuiin saa kolmiulotteisuutta varjoksella, joka tapahtuu komennolla `\shade[<optiot>] <polku>;`. Optioina voi antaa varjostusvärin (tai -värit) ja varjostustavan, joka voi olla `ball`, `axis` tai `radial`.

Varjostaminen `axis`-optiolla tapahtuu antamalla varjostusvärit optioilla joko `top/middle/bottom color=<väri>` tai `left/middle/right color=<väri>`. Näistä yksi tai kaksi voidaan jättää pois, jolloin oletusarvo on valkoinen. Vastaavasti `radial`-varjostaminen tapahtuu antamalla `inner/outer color=<väri>`. Optio `ball` tulee käyttöön optiolla `ball color=<väri>`.

Esimerkkinä varjostuksista lisäämme aiemmin piirtämiimme kuviin sylinteri, ympyrä ja suorakaide varjostuksia:

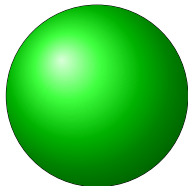
Esimerkkinä varjostuksista lisäämme aiemmin piirtämiimme kuviin sylinteri, ympyrä ja suorakaide varjostuksia:

- 1 `\shade[left color=blue!70!white,middle color=white,right color=blue!50!black] (0,0) – (0,3) arc (180:360:1cm and 0.25cm) – (2,0) arc(0:-180:1cm and 0.3cm);`
`\shade[color=gray] (1,3) circle (1cm and 0.25cm);`



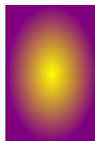
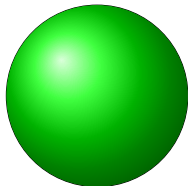
Esimerkkinä varjostuksista lisäämme aiemmin piirtämiimme kuviin sylinteri, ympyrä ja suorakaide varjostuksia:

- 1 `\shade[left color=blue!70!white,middle color=white,right color=blue!50!black] (0,0) – (0,3) arc (180:360:1cm and 0.25cm) – (2,0) arc(0:-180:1cm and 0.3cm);`
`\shade[color=gray] (1,3) circle (1cm and 0.25cm);`
- 2 `\shade[ball color=green] (2,2) circle (2cm);`



Esimerkkinä varjostuksista lisäämme aiemmin piirtämiimme kuviin sylinteri, ympyrä ja suorakaide varjostuksia:

- 1 `\shade[left color=blue!70!white,middle color=white,right color=blue!50!black] (0,0) – (0,3) arc (180:360:1cm and 0.25cm) – (2,0) arc(0:-180:1cm and 0.3cm);`
`\shade[color=gray] (1,3) circle (1cm and 0.25cm);`
- 2 `\shade[ball color=green] (2,2) circle (2cm);`
- 3 `\shade[inner color=yellow,outer color=violet] (-3,-2) rectangle (-1,1);`



Akseleiden nuolenpäiden piirtämiseen ladataan kirjasto
`\usetikzlibrary{arrows}`.

Akseleiden nuolenpäiden piirtämiseen ladataan kirjasto `\usetikzlibrary{arrows}`. Itse akselit piirretään sen jälkeen kuin tavalliset suorat viivat, mutta lisätään optio `->` ja optio haluttaessa nuolenkärjen tyypistä, esim. `triangle 45`.

Akseleiden nuolenpäiden piirtämiseen ladataan kirjasto `\usetikzlibrary{arrows}`. Itse akselit piirretään sen jälkeen kuin tavalliset suorat viivat, mutta lisätään optio `->` ja optio haluttaessa nuolenkärjen tyypistä, esim. `triangle 45`.

Koordinaattimerkkien ja niiden numerointiin tarvitaan hieman ohjelmointia. Jos esimerkiksi halutaan 6 pisteen mittaiset koordinaattimerkit ja numerointi molemmille akselille miinus kolmesta kolmeen, annetaan komentojono

Akseleiden nuolenpäiden piirtämiseen ladataan kirjasto `\usetikzlibrary{arrows}`. Itse akselit piirretään sen jälkeen kuin tavalliset suorat viivat, mutta lisätään optio `->` ja optio haluttaessa nuolenkärjen tyypistä, esim. `triangle 45`.

Koordinaattimerkkien ja niiden numerointiin tarvitaan hieman ohjelmointia. Jos esimerkiksi halutaan 6 pisteen mittaiset koordinaattimerkit ja numerointi molemmille akselille miinus kolmesta kolmeen, annetaan komentojono

```
\foreach \x in {-3,-2,-1,1,2,3}
\draw[shift={(\x,0)}] (0pt,3pt) -- (0pt,-3pt) node[below] {\x};
\draw[>=triangle 45,->] (-3.5,0) -- (3.5,0);
\foreach \y in {-3,-2,-1,1,2,3}
\draw[shift={(0,\y)}] (3pt,0pt) -- (-3pt, 0pt) node[left] {\y};
\draw[>=triangle 45,->] (0,-3.5) -- (0,3.5);
```

Seuraava rivi

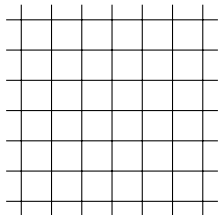
Seuraava rivi

```
\draw (-3.5,-3.5) grid (3.5,3.5);
```

Seuraava rivi

```
\draw (-3.5,-3.5) grid (3.5,3.5);
```

piirtää ruudukon

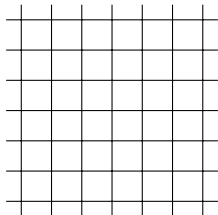


Seuraava rivi

```
\draw (-3.5,-3.5) grid (3.5,3.5);
```

piirtää ruudukon

Oletusaskel ruudukossa on 1, mutta arvoa voi muuttaa optioilla `step=<luku>` tai erikseen `xstep=<luku>`, `ystep=<luku>..`



Seuraava rivi

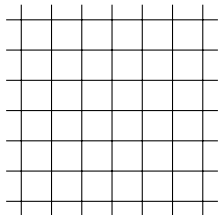
```
\draw (-3.5,-3.5) grid (3.5,3.5);
```

piirtää ruudukon

Oletusaskel ruudukossa on 1, mutta arvoa voi muuttaa optioilla `step=<luku>` tai erikseen `xstep=<luku>`, `ystep=<luku>..`

Eismerkiksi rivi

```
\draw[xstep=0.5,ystep=0.2,color=blue] (-3.5,-3.5) grid (3.5,3.5);
```



Seuraava rivi

```
\draw (-3.5,-3.5) grid (3.5,3.5);
```

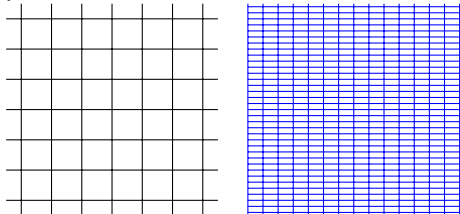
piirtää ruudukon

Oletusaskel ruudukossa on 1, mutta arvoa voi muuttaa optioilla `step=<luku>` tai erikseen `xstep=<luku>`, `ystep=<luku>`..

Eismerkiksi rivi

```
\draw[xstep=0.5,ystep=0.2,color=blue] (-3.5,-3.5) grid (3.5,3.5);
```

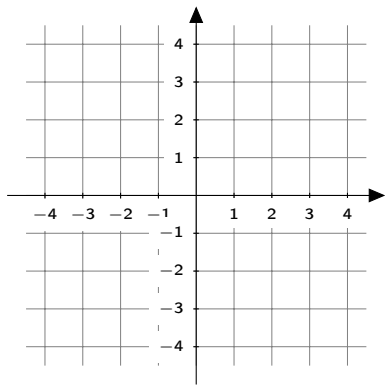
piirtää ruudukon



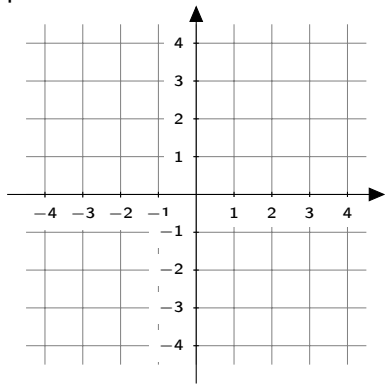
Ruudukkoa täytyy usein “häivyttää”. Ensimmäisessä kuvassa tämä on tehty optioilla `very thin,gray` ja `node:`en on lisätty optio `fill=white`.

Esimerkkejä

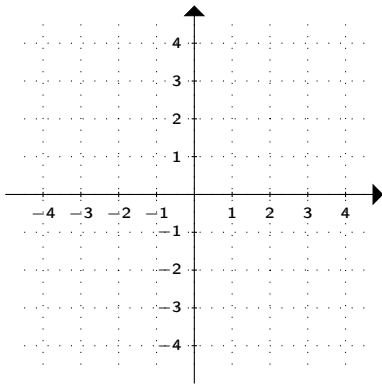
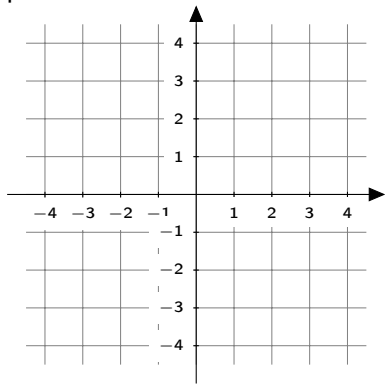
Ruudukkoa täytyy usein “häivyttää”. Ensimmäisessä kuvassa tämä on tehty optioilla `very thin,gray` ja `node`:en on lisätty optio `fill=white`.



Ruudukkoa täytyy usein “häivyttää”. Ensimmäisessä kuvassa tämä on tehty optioilla **very thin,gray** ja **node**:en on lisätty optio **fill=white**. Toisessa option **loosely dotted** johdosta ruudukko on pisteiviivalla.



Ruudukkoa täytyy usein “häivyttää”. Ensimmäisessä kuvassa tämä on tehty optioilla `very thin,gray` ja `node`:en on lisätty optio `fill=white`. Toisessa option `loosely dotted` johdosta ruudukko on pisteiviivalla.



Funktion kuvaajien piirtäminen

Funktion $f(x)$ kuvaajan määrittäminen osaksi jotain polkua tapahtuu toiminnolla

Funktion kuvaajien piirtäminen

Funktion $f(x)$ kuvaajan määrittäminen osaksi jotain polkua tapahtuu toiminnolla

```
plot[domain=<xmin>:<xmax>] (\x,{f(\x)});
```

Funktion kuvaajien piirtäminen

Funktion $f(x)$ kuvaajan määrittäminen osaksi jotain polkua tapahtuu toiminnolla

```
plot[domain=<xmin>:<xmax>] (\x,{f(\x)});
```

Huomaa muuttuja $\backslash x$. TikZ tuntee kaikki alkeisfunktiot, jotka annetaan varsin luonnollisella tavalla.

Funktion kuvaajien piirtäminen

Funktion $f(x)$ kuvaajan määrittäminen osaksi jotain polkua tapahtuu toiminnolla

```
plot[domain=<xmin>:<xmax>] (\x,{f(\x)});
```

Huomaa muuttuja $\backslash x$. TikZ tuntee kaikki alkeisfunktiot, jotka annetaan varsin luonnollisella tavalla.

Taulukossa on esitetty tavallisimmin tarvittavat:

peruslaskutoimitukset	$+$, $-$, $*$, $/$
-----------------------	-----------------------

Funktion kuvaajien piirtäminen

Funktion $f(x)$ kuvaajan määrittäminen osaksi jotain polkua tapahtuu toiminnolla

`plot[domain=<xmin>:<xmax>] (\x,{f(\x)});`

Huomaa muuttuja $\backslash x$. TikZ tuntee kaikki alkeisfunktiot, jotka annetaan varsin luonnollisella tavalla.

Taulukossa on esitetty tavallisimmin tarvitut:

peruslaskutoimitukset	$+, -, *, /$
x potenssiin y	x^y

Funktion kuvaajien piirtäminen

Funktion $f(x)$ kuvaajan määrittäminen osaksi jotain polkua tapahtuu toiminnolla

`plot[domain=<xmin>:<xmax>] (\x,{f(\x)});`

Huomaa muuttuja `\x`. TikZ tuntee kaikki alkeisfunktiot, jotka annetaan varsin luonnollisella tavalla.

Taulukossa on esitetty tavallisimmin tarvitut:

peruslaskutoimitukset	<code>+, -, *, /</code>
x potenssiin y	<code>x^y</code>
neliöjuuri	<code>sqrt(x)</code>

Funktion kuvaajien piirtäminen

Funktion $f(x)$ kuvaajan määrittäminen osaksi jotain polkua tapahtuu toiminnolla

`plot[domain=<xmin>:<xmax>] (\x,{f(\x)});`

Huomaa muuttuja `\x`. TikZ tuntee kaikki alkeisfunktiot, jotka annetaan varsin luonnollisella tavalla.

Taulukossa on esitetty tavallisimmin tarvittavat:

peruslaskutoimitukset	<code>+, -, *, /</code>
x potenssiin y	<code>x^y</code>
neliöjuuri	<code>sqrt(x)</code>
trigonometriset funktiot	<code>sin(x), cos(x), tan(x)</code>

Funktion kuvaajien piirtäminen

Funktion $f(x)$ kuvaajan määrittäminen osaksi jotain polkua tapahtuu toiminnolla

`plot[domain=<xmin>:<xmax>] (\x,{f(\x)});`

Huomaa muuttuja `\x`. TikZ tuntee kaikki alkeisfunktiot, jotka annetaan varsin luonnollisella tavalla.

Taulukossa on esitetty tavallisimmin tarvittavat:

peruslaskutoimitukset	<code>+, -, *, /</code>
x potenssiin y	<code>x^y</code>
neliöjuuri	<code>sqrt(x)</code>
trigonometriset funktiot	<code>sin(x), cos(x), tan(x)</code>
logaritmifunktiot	<code>ln(x), log10(x), log2(x)</code>
π ja e	<code>pi</code> ja <code>e</code>

Funktion kuvaajien piirtäminen

Funktion $f(x)$ kuvaajan määrittäminen osaksi jotain polkua tapahtuu toiminnolla

`plot[domain=<xmin>:<xmax>] (\x,{f(\x)});`

Huomaa muuttuja $\backslash x$. TikZ tuntee kaikki alkeisfunktiot, jotka annetaan varsin luonnollisella tavalla.

Taulukossa on esitetty tavallisimmin tarvittavat:

peruslaskutoimitukset	$+, -, *, /$
x potenssiin y	x^y
neliöjuuri	$\text{sqrt}(x)$
trigonometriset funktiot	$\sin(x), \cos(x), \tan(x)$
logaritmifunktiot	$\ln(x), \log_{10}(x), \log_2(x)$
π ja e	pi ja e
eksponenttifunktio	$\text{exp}(x)$

Funktion kuvaajien piirtäminen

Funktion $f(x)$ kuvaajan määrittäminen osaksi jotain polkua tapahtuu toiminnolla

`plot[domain=<xmin>:<xmax>] (\x,{f(\x)});`

Huomaa muuttuja $\backslash x$. TikZ tuntee kaikki alkeisfunktiot, jotka annetaan varsin luonnollisella tavalla.

Taulukossa on esitetty tavallisimmin tarvittavat:

peruslaskutoimitukset	$+, -, *, /$
x potenssiin y	x^y
neliöjuuri	$\text{sqrt}(x)$
trigonometriset funktiot	$\sin(x), \cos(x), \tan(x)$
logaritmifunktiot	$\ln(x), \log_{10}(x), \log_2(x)$
π ja e	pi ja e
eksponenttifunktio	$\text{exp}(x)$
radiaanien muuttaminen asteiksi	r

Funktion kuvaajien piirtäminen

Funktion $f(x)$ kuvaajan määrittäminen osaksi jotain polkua tapahtuu toiminnolla

`plot[domain=<xmin>:<xmax>] (\x,{f(\x)});`

Huomaa muuttuja $\backslash x$. TikZ tuntee kaikki alkeisfunktiot, jotka annetaan varsin luonnollisella tavalla.

Taulukossa on esitetty tavallisimmin tarvittavat:

peruslaskutoimitukset	$+, -, *, /$
x potenssiin y	x^y
neliöjuuri	$\text{sqrt}(x)$
trigonometriset funktiot	$\sin(x), \cos(x), \tan(x)$
logaritmifunktiot	$\ln(x), \log_{10}(x), \log_2(x)$
π ja e	pi ja e
eksponenttifunktio	$\text{exp}(x)$
radiaanien muuttaminen asteiksi	r
itseisarvo	$\text{abs}(x)$

Komennot

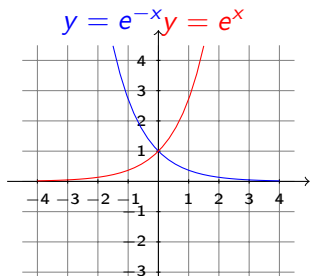
```
\draw[color=blue] (-1.5,4.5)
node[above] { $y=e^{-x}$ }
plot[domain=-1.5:4]
(\x,{ exp(-\x) });
\draw[color=red]
plot[domain=-4:1.5]
(\x,{ exp(\x) })
node[above] { $y=e^x$ };
```

Esimerkkejä

Komennot

```
\draw[color=blue] (-1.5,4.5)
node[above] { $y=e^{-x}$ }
plot[domain=-1.5:4]
(\x,{exp(-\x)});
\draw[color=red]
plot[domain=-4:1.5]
(\x,{exp(\x)})
node[above] { $y=e^x$ };
```

Tuottavat kuvan

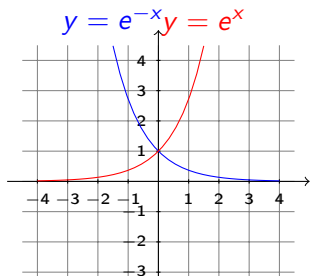


Esimerkkejä

Komennot

```
\draw[color=blue] (-1.5,4.5)
node[above] { $y=e^{-x}$ }
plot[domain=-1.5:4]
(\x,{exp(-\x)});
\draw[color=red]
plot[domain=-4:1.5]
(\x,{exp(\x)})
node[above] { $y=e^x$};
```

Tuottavat kuvan



ja komennot

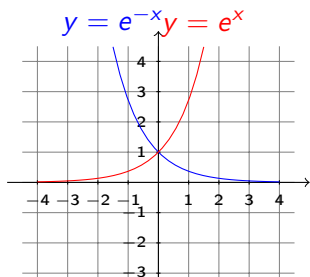
```
\draw[color=green]
plot[domain=-4:1] (\x,{\x})
plot[domain=1:4]
(\x,{(\x)^2-4*\x+4});
```

Esimerkkejä

Komennot

```
\draw[color=blue] (-1.5,4.5)
node[above] { $y=e^{-x}$ }
plot[domain=-1.5:4]
(\x,{exp(-\x)});
\draw[color=red]
plot[domain=-4:1.5]
(\x,{exp(\x)})
node[above] { $y=e^x$ };
```

Tuottavat kuvan

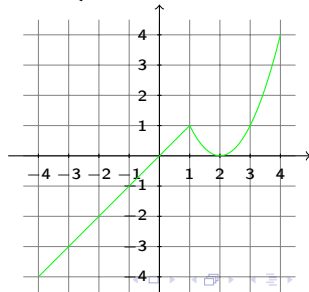


ja komennot

```
\draw[color=green]
plot[domain=-4:1] (\x,{\x})
plot[domain=1:4]
(\x,{(\x)^2-4*\x+4});
```

piirtävät kuvaajan

$$y = \begin{cases} x, & \text{kun } x < 1 \\ x^2 - 4x + 4, & \text{kun } x \geq 1 \end{cases}$$



Geometrinen kuvien piirtäminen

Geometriassa tarvitaan usein kuvia, joissa on paljon nimettyjä pisteitä. TikZ tarjoaa tähän komennon

```
\coordinate (<name>) at <piste>;
```

Geometrinen kuvien piirtäminen

Geometriassa tarvitaan usein kuvia, joissa on paljon nimettyjä pisteitä. TikZ tarjoaa tähän komennon

```
\coordinate (<name>) at <piste>;
```

Esimerkiksi komentojono

```
\coordinate[label=above:{$ A$}](A) at (1,1);
```

```
\draw (A) node{$ \bullet$};
```

A
•

määrittelee pisteen A, kirjoittaa nimen A pisteen yläpuolelle ja merkitsee sen täplällä.

Geometristen kuvien piirtäminen

Geometriassa tarvitaan usein kuvia, joissa on paljon nimettyjä pisteitä. TikZ tarjoaa tähän komennon

```
\coordinate (<name>) at <piste>;
```

Esimerkiksi komentojono

```
\coordinate[label=above:{$ A$}](A) at (1,1);
```

```
\draw (A) node{$ \bullet$};
```

A
•

määrittelee pisteen A , kirjoittaa nimen A pisteen yläpuolelle ja merkitsee sen täplällä. Huomaa, että coordinate-komento yksinään ei piirrä mitään.

Geometristen kuvien piirtäminen

Geometriassa tarvitaan usein kuvia, joissa on paljon nimettyjä pisteitä. TikZ tarjoaa tähän komennon

```
\coordinate (<name>) at <piste>;
```

Esimerkiksi komentojono

```
\coordinate[label=above:{$ A$}](A) at (1,1);
```

```
\draw (A) node{$ \bullet$};
```

A
•

määrittelee pisteen A , kirjoittaa nimen A pisteen yläpuolelle ja merkitsee sen täplällä. Huomaa, että coordinate-komento yksinään ei piirrä mitään.

Kun useita pisteitä on määritelty, voidaan pisteiden nimiä käyttää piirtokomennoissa samoin kuin koordinaattien avulla ilmoitettuja pisteitä. Pisteen nimi laitetaan aina sulkuihin.

Geometristen kuvien piirtäminen

Geometriassa tarvitaan usein kuvia, joissa on paljon nimettyjä pisteitä. TikZ tarjoaa tähän komennon

```
\coordinate (<name>) at <piste>;
```

Esimerkiksi komentojono

```
\coordinate[label=above:{$ A$}](A) at (1,1);
```

```
\draw (A) node{$ \bullet$};
```

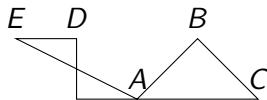


määrittelee pisteen A , kirjoittaa nimen A pisteen yläpuolelle ja merkitsee sen täplällä. Huomaa, että coordinate-komento yksinään ei piirrä mitään.

Kun useita pisteitä on määritelty, voidaan pisteiden nimiä käyttää piirtokomennoissa samoin kuin koordinaattien avulla ilmoitettuja pisteitä. Pisteen nimi laitetaan aina sulkuihin.

Viereinen kuva on piirretty määrittelemällä ylläolevalla tavalla pisteet A , B , C , D ja E ja sen jälkeen annettu komento

```
\draw (A) -- (B) -- (C) -- (D) -- (E) -- cycle;
```



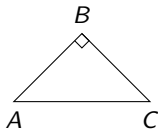
Geometrinen kuvien piirtäminen

Tavallinen ongelma geometrisissa kuvissa on suorien kulmien merkitseminen. Tähänkin TikZ tarjoaa näppärän apuneuvon.

Geometrinen kuvien piirtäminen

Tavallinen ongelma geometrisissa kuvissa on suorien kulmien merkitseminen. Tähänkin TikZ tarjoaa näppärän apuneuvon.

Viereisessä kuvassa on määritelty pisteet A , B ja C , piirretty kolmio ABC ja sen jälkeen annettu komento

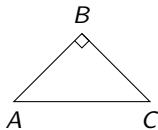


```
\draw (B) -- ++(3pt,-3pt) -- ++ (-3pt,-3pt) -- ++(-3pt,3pt);
```

Geometristen kuvien piirtäminen

Tavallinen ongelma geometrisissa kuvissa on suorien kulmien merkitseminen. Tähänkin TikZ tarjoaa näppärän apuneuvon.

Viereisessä kuvassa on määritelty pisteet A , B ja C , piirretty kolmio ABC ja sen jälkeen annettu komento



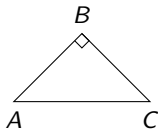
```
\draw (B) -- ++(3pt,-3pt) -- ++ (-3pt,-3pt) -- ++(-3pt,3pt);
```

Idea on siis se, että toiminto `++(<mitta1>,<mitta2>)` lisää nykyisen pisteen x -koordinaattiin $\langle\text{mitta1}\rangle$:n verran ja y -koordinaattiin $\langle\text{mitta2}\rangle$:n verran ja siirtää nykyisen pisteen uuteen pisteeseen.

Geometristen kuvien piirtäminen

Tavallinen ongelma geometrisissa kuvissa on suorien kulmien merkitseminen. Tähänkin TikZ tarjoaa näppärän apuneuvon.

Viereisessä kuvassa on määritelty pisteet A , B ja C , piirretty kolmio ABC ja sen jälkeen annettu komento



```
\draw (B) -- ++(3pt,-3pt) -- ++(-3pt,-3pt) -- ++(-3pt,3pt);
```

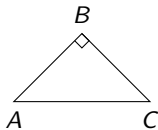
Idea on siis se, että toiminto `++(<mitta1>,<mitta2>)` lisää nykyisen pisteen x -koordinaattiin $\langle\text{mitta1}\rangle$:n verran ja y -koordinaattiin $\langle\text{mitta2}\rangle$:n verran ja siirtää nykyisen pisteen uuteen pisteeseen.

Mikäli janojen koordinaatit ovat hankalia tai niitä ei edes tunneta, suoran kulman merkki saadaan piirrettyä myöhemmin esiteltävien laskentaominaisuuksien avulla kohtuullisen helpolla.

Geometristen kuvien piirtäminen

Tavallinen ongelma geometrisissa kuvissa on suorien kulmien merkitseminen. Tähänkin TikZ tarjoaa näppärän apuneuvon.

Viereisessä kuvassa on määritelty pisteet A , B ja C , piirretty kolmio ABC ja sen jälkeen annettu komento



```
\draw (B) -- ++(3pt,-3pt) -- ++(-3pt,-3pt) -- ++(-3pt,3pt);
```

Idea on siis se, että toiminto `++(<mitta1>,<mitta2>)` lisää nykyisen pisteen x -koordinaattiin $\langle\text{mitta1}\rangle$:n verran ja y -koordinaattiin $\langle\text{mitta2}\rangle$:n verran ja siirtää nykyisen pisteen uuteen pisteeseen.

Mikäli janojen koordinaatit ovat hankalia tai niitä ei edes tunneta, suoran kulman merkki saadaan piirrettyä myöhemmin esiteltävien laskentaominaisuuksien avulla kohtuullisen helpolla.

Muiden kulmakaarten tekemisessä joutuu usein ottamaan kynän ja laskimen kauniisti käteen ja laskemaan kulman suuruuden ja pisteen, josta kaari piirretään.

NAPAKOORDINAATIT

Edelliseen ongelmaa tarjoaa ratkaisun myös napakoordinaattien käyttäminen, jolloin kulma on automaattisesti tunnettu.

Napakoordinaatti määritellään `\coordinate`-komennolla kuten tavallinenkin piste antamalla koordinaatit muodossa

`(<kulma>:<säde>)`. Esimerkiksi `(50:4)` määrittää pisteen, johon päästään kulkemalla 50 asteen kulmassa x-akseliin nähden 4 pituusyksikköä.

Edelliseen ongelmaa tarjoaa ratkaisun myös napakoordinaattien käyttäminen, jolloin kulma on automaattisesti tunnettu.

Napakoordinaatti määritellään `\coordinate`-komennolla kuten tavallinenkin piste antamalla koordinaatit muodossa

`(<kulma>:<säde>)`. Esimerkiksi `(50:4)` määrittää pisteen, johon päästään kulkemalla 50 asteen kulmassa x-akseliin nähden 4 pituusyksikköä.

Tavallisia koordinaatteja ja napakoordinaatteja voi käyttää sekaisin ja myös napakoordinaatteja voi lisätä `++`-toiminnolla toisiin jo määriteltyihin koordinaatteihin.

NAPAKOORDINAATIT

Edelliseen ongelmaa tarjoaa ratkaisun myös napakoordinaattien käyttäminen, jolloin kulma on automaattisesti tunnettu.

Napakoordinaatti määritellään `\coordinate`-komennolla kuten tavallinenkin piste antamalla koordinaatit muodossa `(<kulma>:<säde>)`. Esimerkiksi `(50:4)` määrittää pisteen, johon päästään kulkemalla 50 asteen kulmassa x-akseliin nähden 4 pituusyksikköä.

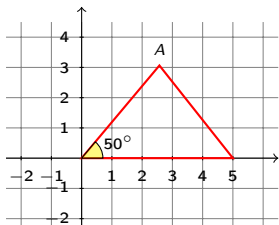
Tavallisia koordinaatteja ja napakoordinaatteja voi käyttää sekaisin ja myös napakoordinaatteja voi lisätä `++`-toiminnolla toisiin jo määriteltyihin koordinaatteihin.

Koodi

```
\coordinate[label=above: {$ A$}](A) at (50:4);
```

```
\draw[red,thick](A)--(0,0)--(5,0)--cycle;
```

piirtää viereisen kolmion.



Geometrinen kuvien piirtäminen

TikZ tarjoaa myös laskennallisia toimintoja. Niitä varten täytyy alussa ladata kirjasto `calc` komennolla `\usetikzlibrary{calc}`.

Geometrinen kuvien piirtäminen

TikZ tarjoaa myös laskennallisia toimintoja. Niitä varten täytyy alussa ladata kirjasto `calc` komennolla `\usetikzlibrary{calc}`.

Toiminto `rand` arpoo luvun väliltä $[-1, 1]$. Esimerkiksi `\coordinate (A) at (1+0.2*rand,1+0.2*rand);` määrittelee pisteen, jonka koordinaatit ovat välillä $[0.8, 1.2]$.

Geometrinen kuvien piirtäminen

TikZ tarjoaa myös laskennallisia toimintoja. Niitä varten täytyy alussa ladata kirjasto `calc` komennolla `\usetikzlibrary{calc}`.
Toiminto `rand` arpoo luvun väliltä $[-1, 1]$. Esimerkiksi `\coordinate (A) at (1+0.2*rand,1+0.2*rand);` määrittelee pisteen, jonka koordinaatit ovat välillä $[0.8, 1.2]$. Tämä toiminnon avulla voi välttää vaaraa tehdä liian säännöllisiä geometrisia kuvia, jolloin oppilaat helposti katsovat tuloksen kuvasta. Koska `rand` arvotaan joka kerta uudelleen, voi sulkemalla ja lataamalla kuvan uudelleen demostroida oppilaille sitä, että menetelmä toimii kaikissa tapauksissa.

Geometristen kuvien piirtäminen

TikZ tarjoaa myös laskennallisia toimintoja. Niitä varten täytyy alussa ladata kirjasto `calc` komennolla `\usetikzlibrary{calc}`. Toiminto `rand` arpoo luvun väliltä $[-1, 1]$. Esimerkiksi `\coordinate (A) at (1+0.2*rand,1+0.2*rand);` määrittelee pisteen, jonka koordinaatit ovat välillä $[0.8, 1.2]$. Tämä toiminnon avulla voi välttää vaaraa tehdä liian säännöllisiä geometrisia kuvia, jolloin oppilaat helposti katsovat tuloksen kuvasta. Koska `rand` arvotaan joka kerta uudelleen, voi sulkemalla ja lataamalla kuvan uudelleen demostroida oppilaille sitä, että menetelmä toimii kaikissa tapauksissa.

Käytettäessä satunnaisia arvoja törmätään ongelmaan, jos tarvitaan esimerkiksi janan pituutta. Calc-kirjasto tarjoaa tähän `veclen`-funktion, joka laskee vektorin (x, y) (lukiolaisten kielellä $x\vec{i} + y\vec{j}$) pituuden. Myös kiinteissä kuvissa voi joskus olla työlästä laskea janan pituuksia.

Geometristen kuvien piirtäminen

TikZ tarjoaa myös laskennallisia toimintoja. Niitä varten täytyy alussa ladata kirjasto `calc` komennolla `\usetikzlibrary{calc}`. Toiminto `rand` arpoo luvun väliltä $[-1, 1]$. Esimerkiksi `\coordinate (A) at (1+0.2*rand,1+0.2*rand);` määrittelee pisteen, jonka koordinaatit ovat välillä $[0.8, 1.2]$. Tämä toiminnon avulla voi välttää vaaraa tehdä liian säännöllisiä geometrisia kuvia, jolloin oppilaat helposti katsovat tuloksen kuvasta. Koska `rand` arvotaan joka kerta uudelleen, voi sulkemalla ja lataamalla kuvan uudelleen demostroida oppilaille sitä, että menetelmä toimii kaikissa tapauksissa.

Käytettäessä satunnaisia arvoja törmätään ongelmaan, jos tarvitaan esimerkiksi janan pituutta. Calc-kirjasto tarjoaa tähän `veclen`-funktion, joka laskee vektorin (x, y) (lukiolaisten kielellä $x\vec{i} + y\vec{j}$) pituuden. Myös kiinteissä kuvissa voi joskus olla työlästä laskea janan pituuksia. Seuraavassa esimerkissä arvotaan on arvottu jana AB , jossa A :n koordinaatit ovat välillä $[0.8, 1.2]$ ja B :n koordinaatit välillä $[-1.2, -0.8]$.

Geometristen kuvien piirtäminen

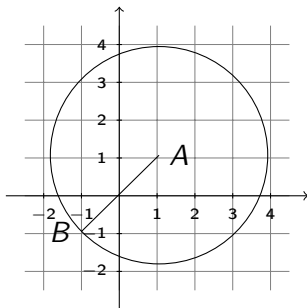
Koodi kuvan vieressä laskee janan pituuden ja piirtää ympyrä, jonka säde on AB .

```
\draw (A) let
```

```
\p1=(B)-(A)
```

```
in circle ({veclen(\x1,\y1)});
```

Tässä $\langle p \rangle$ on vektorin nimi, jolloin $\langle x \rangle$ ja $\langle y \rangle$ ovat vektorin komponentit.



Geometrinen kuvien piirtäminen

Koodi kuvan vieressä laskee janan pituuden ja piirtää ympyrä, jonka säde on AB .

```
\draw (A) let
```

```
\p1=(\$(B)-(A)\$)
```

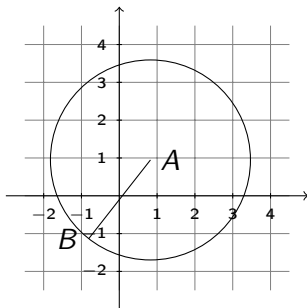
```
in circle ({veclen(\x1,\y1)});
```

Tässä $\langle p \rangle$ on vektorin nimi, jolloin $\langle x \rangle$ ja $\langle y \rangle$ ovat vektorin komponentit.

Toinen yleinen ongelma on geometrinen kuvioiden tai käyrien leikkauspisteiden laskeminen. Siihen on tarjolla

intersections-tominto. Idea on yksinkertainen: nimetään kaksi polkua, esimerkiksi P ja Q , ja sen jälkeen **name**

intersections={of=P and Q} nimeää leikkauspisteet **intersection-1**, **intersection-2**, jne..



Geometrinen kuvien piirtäminen

Koodi kuvan vieressä laskee janan pituuden ja piirtää ympyrä, jonka säde on AB .

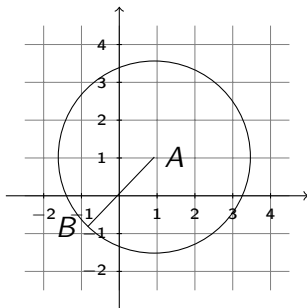
```
\draw (A) let
```

```
\p1=( $(B)-(A)$ )
```

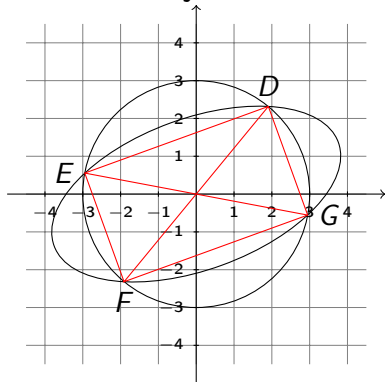
```
in circle ({veclen( $\x1,\y1$ )});
```

Tässä $\langle p \rangle$ on vektorin nimi, jolloin $\langle x \rangle$ ja $\langle y \rangle$ ovat vektorin komponentit.

Toinen yleinen ongelma on geometrinen kuvioiden tai käyrien leikkauspisteiden laskeminen. Siihen on tarjolla intersections-toiminto. Idea on yksinkertainen: nimetään kaksi polkua, esimerkiksi P ja Q , ja sen jälkeen `name intersections={of= P and Q }` nimeää leikkauspisteet `intersection-1`, `intersection-2`, jne.. Seuraavassa esimerkissä piirretään origokeskeiset ympyrä ja ellipsi ja yhdistetään leikkauspisteet janoilla.



Allaolevasta koodista on jätetty pois alkuosa, jossa piirretään koordinaatisto ja ruudukko.



```
\draw[name path=P] (0,0) circle (3);
\draw [name path=Q, rotate=20] (0,0) circle
(4 and 2) ;
\path[name intersections={ of=P and Q }];
\coordinate[label=above: {$ D$ }] (D) at
(intersection-1);
\coordinate[label=left: {$ E$ }] (E) at
(intersection-2);
\coordinate[label=below: {$ F$ }] (F) at
(intersection-3);
\coordinate[label=right: {$ G$ }] (G) at
(intersection-4);
\draw[red] (D) -- (F) -- (G) -- (D) -- (E) --
(F) ;

\draw[red] (E) -- (G);
```