

L<sup>A</sup>T<sub>E</sub>X

8. huhtikuuta 2014

### 3. kerta

#### Kuvat

- Kuvien liittäminen
- Figure-ympäristö
- Piirtäminen (TikZ)

#### Taulukot

- Tabular-ympäristö
- Sarakkeiden yhdistäminen
- Rivien yhdistäminen
- Kelluva table-ympäristö

#### Matriisit

- Matriisiympäristöt
- Lisätietoa

### 4.kerta

#### Viittaaminen kuviin ja taulukoihin

#### Kirjallisuusviitteet

- Lainaukset ja alaviitteet
- BibTeX-järjestelmä
- Viittaustyyli

#### Dokumentin viimeistely

- Sisällysluettelo ja kansilehti
- Dokumenttiluokan asetukset

## Kuvien liittäminen

Kuvien liittämistä varten täytyy ottaa käyttöön `graphicx`-paketti.

PDFLaTeXia käytettäessä sallitut kuvaformaattit ovat `.pdf`, `.jpg` ja `.png`.

Jos liitettävä kuva on samassa kansiossa muiden tiedostojen kanssa, kuvan liittäminen tapahtuu komennolla

```
\includegraphics[asetukset]{kuvatiedoston_nimi}.
```

Tiedostopäätettä ei ole tarpeen kirjoittaa, jos tiedoston nimi ei sisällä välejä.

Valinnaisella argumentilla voidaan skaalata, kiertää ja rajata liitettävää kuvaa.

## Kuvien liittäminen

Työkansioon tallennetun kuvan smile.jpeg saisi näkyviin koodilla

```
\includegraphics{smile}
```



# Kuvan skaalaus

Kuva on aivan liian suuri, joten sitä on syytä skaalata:

```
\includegraphics[scale=0.4]{smile}
```



# Kuvan skaalaus

Skaalaus onnistuu myös määrittelemällä kuvan leveys tai korkeus:

```
\includegraphics[width = 4cm]{smile}
```

```
\includegraphics[height=4cm]{smile}
```



# Kuvan kierto

Kuvaa voi kiertää antamalla valinnaiseksi argumentiksi kiertokulman asteina.

```
\includegraphics[width=4cm,angle=90]{smile}
```

```
\includegraphics[height=4cm,angle=180]{smile}
```



Huomaa, että useammat määritteet erotetaan toisistaan pilkuilla!

## Harjoitus 3.1

Tallenna työkansioosi jokin kuva (tarkkana formaatin kanssa) ja tuo se työhösi sopivasti skaalattuna. Muista ottaa ensin käyttöön paketti `graphicx`!



## Kelluva figure-ympäristö

Käyttämällä pelkästään komentoa `\includegraphics[]{} kuva` tuodaan komennon osoittamaan paikkaan, kuin osaksi tekstiä. Tämä näyttää käytännössä aina pahalta.

On parempi antaa  $\text{\LaTeX}$ in päättää itse mihin kuva sijoitetaan eli tehdä kuvasta *kelluva*. Tämä onnistuu käyttämällä figure-ympäristöä.

# Kelluva kuva

```
\begin{figure}[h!]  
\includegraphics[height=4cm]{smile}  
\caption{Helppo hymyillä}  
\end{figure}
```



Kuva: Helppo hymyillä

## Figure-ympäristön argumentit

Kelluvalle ympäristölle (yllä `figure`) voidaan antaa valinnaisena argumenttina toive kuvan sijoittamisesta. Toiveen voi esittää seuraavilla määritteillä:

<code>h!</code>	tähän
<code>h</code>	suunnilleen tähän
<code>b</code>	sivun alareunaan
<code>t</code>	sivun yläreunaan
<code>p</code>	omalle sivulleen

$\text{\LaTeX}$  ei lupaa noudattaa toivetta, mutta tekee parhaansa.

# Figure-ympäristön argumentit

## Harjoitus 3.2

Tuo työhösi jokin kuva käyttäen figure-ympäristöä ja anna valinnaisena argumenttina toiveesi kuvan sijoittamisesta. Lisää vielä kuvateksti.

## Kuvien piirtäminen

Kuvien liittämisen lisäksi niitä voi myös piirtää suoraan  $\text{\LaTeX}$ illa. Yksinkertaisimmissa tapauksissa tämä onnistuu helposti, monimutkaisempien kuvien tuottaminen vaatii harjaantumista.

Kuvien piirtäminen vie aikaa, mutta jälki on sen mukaista eikä erillisiä kuvatiedostoja tarvitse säilyttää.

Kuvien piirtämistä varten on tarjolla muutamia paketteja, joista erityisesti `TikZ` on syytä mainita. `TikZ`llä kuvien piirtäminen tapahtuu ikään kuin pikseli kerrallaan, joten tarkkuus on verraton.

Geogebra osaa kääntää sillä luodut kuvat `tikz`-koodiksi, mikä helpottaa työtä valtavasti.

Muita kuvien tuottamiseen tarkoitettuja paketteja ovat `picture`, `xy-Pic` ja `Asymptote`.

## Harjoitus 3.3

Piirrä Geogebralla jokin yksinkertainen kuva (vältä funktioita). Rajaa kuva sopivasti ja valitse File > Export > Graphics View as PGF/TikZ. Luo koodi ja kopioi siitä tarvittavat osat tiedostoosi. Tarvittavia osia ovat esittelyosan komennot `\usepackage...`, `\usetikzlibrary...`, mahdolliset värien määrittelyt `\definecolor...` ja varsinainen kuva, eli `\begin{tikzpicture}...\end{tikzpicture}`.

## Harjoitus 3.4

Kopioi edellisessä harjoituksessa luomasi kuvan koodi ja tee kuvaan joitakin muutoksia pelkästään koodia muuttamalla. Voit esimerkiksi vaihtaa jonkin pisteen paikkaa tai piirtää kokonaan uutta. Tee uudesta kuvasta kelluva ja keksi jokin kuvateksi.

# Taulukot

Taulukoiden rakentaminen  $\LaTeX$ issa on oma taiteenlajinsa. Mahdollisuudet ovat rajattomat, mutta perusteiden opettelu vaatii hieman energiaa.

Perinteisin tapa on käyttää `tabular`-ympäristöä. Tällöin taulukon sisältö kirjoitetaan komentojen `\begin{tabular}[]{}{}` ja `\end{tabular}` väliin.

Ympäristöllä on pakollinen argumentti, jolla määritellään taulukon asetukset, eli miten kunkin sarakkeen sisältö tasataan ja millaisia pystyviivoja käytetään.

# Tabular-ympäristö

Tabular-ympäristön pakollinen argumentti on jono seuraavia merkkejä:

Merkki	Toiminto
r	sarakkeen sisällön tasaus oikealle
l	sarakkeen sisällön tasaus vasemmalle
c	sarakkeen sisällön keskitys
	pystyviivan paikka
	kaksi pystyviivaa
p{ 'pituus' }	pituus, jonka jälkeen rivi katkaistaan

Esimerkiksi komento `\begin{tabular}{r|cc}` aloittaisi 3-sarakkeisen taulukon, jolla olisi pystyviiva 1. ja 2. sarakkeen välillä. Sarakkeen 1 sisältö tasattaisiin oikealle, kahden muun sisältö keskitettäisiin.



# Tabular-ympäristö

Taulukon sisältö kirjoitetaan rivi kerrallaan. Jokaisella rivillä merkki `&` erottaa sarakkeet toisistaan, komennolla `\\` siirrytään seuraavalle riville. Nämä ja muut taulukon sisäiset komennot ovat seuraavassa:

Komento	Toiminto
<code>&amp;</code>	sarakeenvaihto
<code>\\</code>	rivinvaihto
<code>\hline</code>	vaakaviiva
<code>\newline</code>	rivinvaihto sarakkeen sisällä
<code>\cline{i-j}</code>	vaakaviiva sarakkeiden i ja j välillä

# Tabular-ympäristö

## Esimerkiksi rivi

```
\begin{tabular}{r|cc}
 & Tytöt & Pojat\\
\hline
Sinisilmäiset & 4 & 2 \\
Ruskeasilmäiset & 3 & 5 \\
Vihreäsilmäiset & 8 & 8\\
\end{tabular}
```

luo seuraavan taulukon:

	Tytöt	Pojat
Sinisilmäiset	4	2
Ruskeasilmäiset	3	5
Vihreäsilmäiset	8	8

Huomaa, ettei rivien lukumäärää tarvitse erikseen kertoa  $\LaTeX$ ille.

# Tabular-ympäristö

## Harjoitus 3.5

Laadi jokin 3-sarakkeinen taulukko, jossa on ainakin kaksi riviä.

## Harjoitus 3.6

Luo toinen 3-sarakkeinen taulukko (esim. kopio edellisestä), joka sisältää

- ▶ pystyviivan
- ▶ kaksinkertaisen pystyviivan
- ▶ vaakaviivan
- ▶ osittaisen vaakaviivan kahden sarakkeen välillä

# Sarakkeiden yhdistäminen

Sarakkeiden yhdistäminen onnistuu komennolla `\multicolumn{}{}{}`. Pakollisista argumenteista

- ▶ ensimmäinen on yhdistettävien solujen lukumäärä
- ▶ toinen on yhdistämällä saadun sarakkeen tasaus
- ▶ kolmas on yhdistämällä saadun sarakkeen sisältö

Komento `\multicolumn` toimii sellaisenaan eikä tarvitse lisäpaketteja.

# Sarakkeiden yhdistäminen

```
\begin{tabular}{|c|c|c|}  
\hline  
\multicolumn{3}{|c|}{3 yhdistettyä saraketta}\\  
\hline  
Sarake & \multicolumn{2}{|c|}{2 yhdistettyä saraketta}\\  
\hline  
Sarake1 & Sarake2 & Sarake3\\  
\hline  
\end{tabular}
```

3 yhdistettyä saraketta		
Sarake	2 yhdistettyä saraketta	
Sarake1	Sarake2	Sarake3

# Sarakkeiden yhdistäminen

## Harjoitus 3.7

Luo seuraava taulukko:

Päiväpetolintuja		
<i>Nimitys</i>	<i>Suku</i>	<i>Laji</i>
Kanahaukka	Accipiter	gentilis
Hiirihaukka	Buteo	buteo
Tuulihaukka	Falco	columbarius
Nuolihaukka	Falco	subbuteo

## Rivien yhdistäminen

Rivien yhdistämistä varten tarvitaan paketti `multirow`. Tämän käyttöönottamisen jälkeen rivien yhdistäminen (sarakkeen sisällä) onnistuu komennolla `\multirow{}{}{}`. Pakollisista argumenteista

- ▶ ensimmäinen on yhdistettävien solujen lukumäärä
- ▶ toinen on yhdistämällä saadun rivin leveys (\* jättää asian  $\LaTeX$ in huoleksi)
- ▶ kolmas on yhdistämällä saadun rivin sisältö

Rivin leveyttä ei useinkaan kannata itse valita, ellei ole varma siitä mitä haluaa tehdä.

Huomaa, että `\multirow` toimii kuten `\multicolumn`, mutta keskimäinen argumentti on eri tarkoitusta varten.

# Rivien yhdistäminen

```
\begin{tabular}{|c|c|c|}
\hline
\multirow{3}{*}{Kolme riviä} & Solu1 & Solu 2\\
\cline{2-3}
& \multirow{2}{*}{Kaksi riviä} & Solu 3\\
\cline{3-3}
& & Solu 4\\
\hline
\end{tabular}
```

Kolme riviä	Solu1	Solu 2
	Kaksi riviä	Solu 3
		Solu 4



# Rivien yhdistäminen

## Harjoitus 3.8

Luo seuraava taulukko:

<i>Heimo</i>	<i>Nimitys</i>	<i>Suku</i>	<i>Laji</i>
Haukat	Kanahaukka	Accipiter	gentilis
	Hiirihaukka	Buteo	buteo

# Sarakkeiden ja rivien yhdistäminen

Sarakkeita ja rivejä voi yhdistää samassa taulukossa:

```
\begin{tabular}{|l|l|l|l|}  
\hline  
\multicolumn{4}{|c|}{\Large Päiväpetolintuja}\\  
\hline  
\textit{Heimo} & \textit{Nimitys} & \textit{Suku} & \textit{Laji}\\  
\multirow{2}{*}{Haukat} & Kanahaukka & Accipiter & gentilis\\  
& Hiirihaukka & Buteo & buteo\\  
\hline  
\multirow{2}{*}{Jalohaukat} & Tuulihaukka & Falco & columbarius\\  
& Nuolihaukka & Falco & subbuteo\\  
\hline  
\end{tabular}
```

Päiväpetolintuja			
<i>Heimo</i>	<i>Nimitys</i>	<i>Suku</i>	<i>Laji</i>
Haukat	Kanahaukka	Accipiter	gentilis
	Hiirihaukka	Buteo	buteo
Jalohaukat	Tuulihaukka	Falco	columbarius
	Nuolihaukka	Falco	subbuteo

Taulukoiden rakentamisessa lähes mikä tahansa on mahdollista.  
Kirjasta <http://en.wikibooks.org/wiki/LaTeX/Tables> voi etsiä apua monimutkaisempia toteutuksia varten.

## Kelluva table-ympäristö

Komento `\begin{tabular}{...}... \end{tabular}` luo talukon komennon osoittamaan paikkaan, kuin osaksi tekstiä. Tämä näyttää käytännössä aina pahalta.

On parempi antaa  $\text{\LaTeX}$ in päättää itse mihin taulukko sijoitetaan eli tehdä siitä *kelluva*. Tämä onnistuu sijoittamalla taulukko `table`-ympäristöön.

```
\begin{table}
\begin{tabular}{...}
...
\end{tabular}
\end{table}
```

## Table-ympäristön argumentit

Kelluvalle ympäristölle (yllä `table`) annetaan valinnaisena argumenttina toive taulukon sijoittamisesta. Toiveen voi esittää seuraavilla määritteillä:

<code>h!</code>	tähän
<code>h</code>	suunnilleen tähän
<code>b</code>	sivun alareunaan
<code>t</code>	sivun yläreunaan
<code>p</code>	omalle sivulleen

Toive ei ole  $\text{\LaTeX}$ in tärkeysjärjestyksessä korkeimmalla, joten se ei aina toteudu.

## Harjoitus 3.9

Tee tehtävässä 3.8 luomastasi taulukosta kelluva ja anna sille jokin nimi. Kokeile erilaisia sijoitteluvaihtoehtoja. Lisää koodiisi kommenttirivi, jossa kerrot valintasi.

## Matematiikkatilan taulukot

$\LaTeX$  olettaa tabular-ympäristön sisältävän tavallista tekstiä. Taulukoituja matemaattisia ilmaisuja varten on oma ympäristönsä array. Sitä käytetään kuten tabular-ympäristöä, mutta se täytyy sijoittaa matematiikkatilaan.

```
\(  
\begin{array}{r|c}  
x & x^2+1\\  
\hline  
-1 & 2\\  
0 & 1\\  
1 & 2\\  
\end{array}  
\)
```

$x$	$x^2 + 1$
-1	2
0	1
1	2

## Array-ympäristö

Array-ympäristöllä voi kirjoittaa esimerkiksi paloittain määritellyn funktion lausekkeen. Tämän voi toteuttaa lyhyemminkin käyttäen ympäristöä `cases`.

```
\[  
f(x) =  
\begin{cases}  
x, \text{ kun } x > 0, \\  
-x, \text{ kun } x \leq 0.  
\end{cases}  
\]
```

$$f(x) = \begin{cases} x, & \text{kun } x > 0, \\ -x, & \text{kun } x \leq 0. \end{cases}$$



# Matriisit

Matriisit voitaisiin luoda käsin käyttäen array-ympäristöä, mutta hieman helpompiakin tapoja on. Esimerkiksi ympäristöllä `pmatrix` matriiseja luotaisiin seuraavasti:

```
\[
\begin{pmatrix}
a & b & c \\
1 & 2 & 3
\end{pmatrix}
\]
```

$$\begin{pmatrix} a & b & c \\ 1 & 2 & 3 \end{pmatrix}$$

Syntaksi on siis samanhenkinen kuin ympäristöllä `array`, mutta sarakkeiden määrää ei tarvitse kertoa erikseen. Virheiden varalta muista, että

- ▶ sarakkeet erotetaan toisistaan komennolla `&`
- ▶ rivi vaihdetaan komennolla `\\`

## Harjoitus 3.10

Luo jokin vähintään  $2 \times 3$ -matriisi käyttäen ympäristöä `pmatrix`, kuten yllä. Tee sitten matriisistasi muutama kopio ja kokeile ympäristöjä `matrix`, `bmatrix` ja `vmatrix`. (Kunakin kohdalle kannattaa kirjoittaa kommentti tulostuvan matriisin tyylistä.)

# Matriisit

Vaikka ympäristöt `pmatrix` (`matrix`, `bmatrix`, `vmatrix`) pohjautuvat `array`-ympäristöön, ei sarakkeiden tasaukseen voi lähtökohtaisesti vaikuttaa.

Paketti `mathtools` tarjoaa vastaavat ympäristöt `pmatrix*`, `bmatrix*` jne., joilla on valinnaisena argumenttina sarakkeissa käytetty tasaus:

```
\[
\begin{pmatrix*}[r]
a & b & c \\
-1 & -2 & -3 \\
\end{pmatrix*}
\]
```

$$\begin{pmatrix} a & b & c \\ -1 & -2 & -3 \end{pmatrix}$$

# Matriisit

Esimerkki:

```
\(  
\begin{bmatrix*}[r]  
-1 & 2\\  
0 & 1\\  
1 & -2\\  
\end{bmatrix*}^T =  
\begin{bmatrix*}[r]  
-1 & 0 & 1\\  
2 & 1 & -2  
\end{bmatrix*  
\)
```

$$\begin{bmatrix} -1 & 2 \\ 0 & 1 \\ 1 & -2 \end{bmatrix}^T = \begin{bmatrix} -1 & 0 & 1 \\ 2 & 1 & -2 \end{bmatrix}$$

# Matriisit

## Harjoitus 3.11

Kirjoita seuraavanlainen matriisitoimitus:

$$\begin{bmatrix} 0 & 0 & 1 & a_1 \\ 0 & 1 & 1 & a_2 \\ 1 & 1 & 1 & a_3 \end{bmatrix} \xrightarrow{R_1 \leftrightarrow R_3} \begin{bmatrix} 1 & 1 & 1 & a_3 \\ 0 & 1 & 1 & a_2 \\ 0 & 0 & 1 & a_1 \end{bmatrix}$$

Matriisien sisällön saat päättää vapaasti eikä rivitoimituksen tarvitse mennä oikein. Matriisien välisen merkinnän saat rakennettua komennon `\stackrel{}{\leftrightarrow}` avulla. Yllä on käytetty komentoa

```
\stackrel{}{\leftrightarrow}
\begin{scriptsize}
R_1\leftrightarrow R_3
\end{scriptsize}
}{\longrightarrow}
```

## Lisätietoja matriiseista

Tavalliset matriisiympäristöt `pmatrix` jne. voidaan säätää noudattamaan paremmin `array`-ympäristön syntaksia.

Tämä on tarpeen, jos halutaan valita eri sarakkeisiin erilainen taseaus tai luoda pystyviiva matriisin sisälle.

Syntaksin muuttaminen onnistuu kopioimalla esittelyosaan komennot

```
\makeatletter
\renewcommand*\env@matrix[1][*\c@MaxMatrixCols c]{%
  \hskip -\arraycolsep
  \let\@ifnextchar\new@ifnextchar
  \array{#1}}
\makeatother
```

## Lisätietoa matriiseista

Edellä tehtyjen muutosten jälkeen matriisiympäristöt toimivat kuten ennenkin, mutta lisäksi valinnainen argumentti on käytössä kuten `array`-ympäristöllä:

```
\[
\begin{bmatrix} rrr|r
0 & 0 & 1 & -a_1 \\
0 & 1 & -1 & a_2 \\
1 & 1 & 1 & a_3
\end{bmatrix}
\]
```

$$\left[ \begin{array}{ccc|c} 0 & 0 & 1 & -a_1 \\ 0 & 1 & -1 & a_2 \\ 1 & 1 & 1 & a_3 \end{array} \right]$$

## Viittaaminen kuviin ja taulukoihin

Kelluviksi kirjoitetut kuvat ja taulukot tulevat numeroiduiksi, jolloin niihin viittaaminen onnistuu helposti komennoilla `\label{}` ja `\ref{}`.

### Harjoitus 4.1

Viittaa harjoituksessa 3.9 luomaasi kelluvaan taulukkoon. Käytä komentoja `\ref{}` ja `\pageref{}`.

### Harjoitus 4.2

Sama kuin edellinen harjoitus, mutta viittaa harjoituksessa 3.4 luomaasi kelluvaan kuvaan.



## Lainaukset ja alaviitteet

Suomalaisittain käytettävät lainausmerkit saa kahdella peräkkäisellä ' merkillä:

```
Matti sanoi: ''Onpa outoa!''
```

tulostaa

```
Matti sanoi: "Onpa outoa!"
```

Lainauksia varten on olemassa myös omia ympäristöjään, kuten `verse`, `quote` ja `quotation`.

Alaviitteen voi luoda komennolla `\footnote{teksti}`. Komento kirjoitetaan siihen kohtaan, johon alaviitteen merkki halutaan.

### Harjoitus 4.3

Lisää dokumenttiisi jokin lainaus käyttäen `quote-`, `\quotation-` tai `verse-` ympäristöä.

### Harjoitus 4.4

Lisää työhösi alaviite komennolla `\footnote{}`.

# Kirjallisuusviitteet

$\LaTeX$ issa kirjallisuusviittaukset kannattaa hoitaa *BibTeX*-järjestelmän avulla.

Tällöin jokaista lähdeosta kohden luodaan erillinen `.bib`-tiedosto, joka sisältää teoksen tiedot.

Tiedostot on yksinkertaisinta tallentaa samaan kansioon `.tex`-tiedoston kanssa.

Viittaaminen tapahtuu komennolla `\cite{tunnus}`, jossa `tunnus` on eräs teoksen `.bib`-tiedostosta löytyvä merkkijono.

## .bib-tiedostot

.bib-tiedoston voi luoda millä tahansa tekstieditorilla, erityisen hyvin Texmakerilla.

Tiedoston sisältö voisi olla esimerkiksi seuraava:

```
@book{kemper,  
title={A Course in Commutative Algebra},  
author={Kemper, G.},  
isbn={9783642035456},  
series={Graduate Texts in Mathematics},  
url={http://books.google.fi/books?id=8kxlj48DWM4C},  
year={2010},  
publisher={Springer Berlin Heidelberg}  
}
```

## .bib-tiedostot

### Harjoitus 4.5

Kopioi edellisen esimerkin sisältö uuteen tiedostoon ja tallenna se nimellä `Kemper.bib` työkansioosi.

## .bib-tiedoston sisältö

- ▶ Ensimmäisellä komennolla (`@book`, `@article`, `@unpublished` jne) kerrotaan, minkä tyyppisestä teoksesta on kyse.
- ▶ Aaltosulkeisiin ennen ensimmäistä pilkkua tuleva merkkijono on se tunnus, jolla teokseen viitataan komennolla `\cite{tunnus}`.
- ▶ Tunnuksen voi valita vapaasti (ilman ääkkösiä ja erikoismerkkejä) eikä se tule näkyviin mihinkään. Esimerkiksi yllä tunnus on `kemper`, joten teokseen viitattaisiin komennolla `\cite{kemper}`.
- ▶ Muut kentät (ja niiden pakollisuus/valinnaisuus) määräytyvät teoksen tyyppin mukaan.

## Luokat ja .bib-tiedoston sisältö

Kun tiedostoja luodaan käsin, täytyy selvittää mitkä kentät ovat pakollisia ja mitkä valinnaisia.

Esimerkiksi taulukosta [http://en.wikibooks.org/wiki/LaTeX/Bibliography\\_Management#Entry\\_and\\_field\\_types\\_in\\_.bib\\_files](http://en.wikibooks.org/wiki/LaTeX/Bibliography_Management#Entry_and_field_types_in_.bib_files) selviää, että kirjalle (@book) pakollisia kenttiä ovat title ja author, muut valinnaisia.

Lisätietoa käytettävistä teosluokista löytyy esimerkiksi sivulta [http://en.wikibooks.org/wiki/LaTeX/Bibliography\\_Management#Standard\\_templates](http://en.wikibooks.org/wiki/LaTeX/Bibliography_Management#Standard_templates).

Käytännössä jokaista kuviteltavissa olevaa lähdeosta varten löytyy jokin sopiva luokka (ja tarpeen tullen sellaisen voi luoda itsekin).

## Harjoitus 4.6

Luo BibTeX-tiedosto kirjalle, jonka nimi on Topologia I, kirjoittaja Jussi Väisälä, julkaisija Limes ry ja painovuosi 2000. Valitse viittaustunnukseksi topo1 ja tallenna tiedosto työkansioosi nimellä Vaisala.bib.



## .bib-tiedostot Internetistä

Kirjojen teostietoja löytää BibTeX-muodossa varsin suurella todennäköisyydellä Google Books-palvelusta.

Kuhunkin teokseen liittyvän Tietoja teoksesta-sivun alalaidassa on kohta Vie sitaatti ja sen vieressä painike BibTeX, josta tiedoston voi ladata.

Tiedosto kannattaa avata Texmakerillä, tehdä mahdolliset muutokset ja tallentaa haluttuun kansioon.

Käytännössä ainakin viittaustunnus on syytä vaihtaa helpommaksi.

# .bib-tiedostot Internetistä

## Harjoitus 4.7

Etsi Google Books-palvelusta Walter Rudinin kirjan Complex Analysis (kirjan versiolla ei väliä) tiedot BibTeX-muodossa. Avaa tiedosto Texmakerilla ja valitse viittaustunnus haluamaksesi. Tallenna tiedosto nimellä `Rudin.bib` työkansioosi. (Huomaa, että palvelu tarjoaa tiedostot muodossa `.bibtex`.)

## Kirjallisuusluettelon luominen

Kun tarvittavat .bib-tiedostot on tallennettu koneelle, on syytä luoda kirjallisuusluettelo. Kirjallisuusluetteloon tulee näkyviin ne teokset, joiden tiedot ovat olemassa ja joihin on viitattu ainakin kerran.

Luettelo luodaan työn loppuosaan komennolla

```
\bibliographystyle{luettelon tyyli}  
\bibliography{teos1,teos2,teos3,...}
```

Koodissa teos1 on tiedoston teos1.bib nimi ilman tiedostopäätettä. Huomaa myös, ettei pilkun jälkeen sallita välilyöntiä.

Edellisen koodin toimimisen edellytyksenä on, että tiedostot teos1.bib jne. ovat samassa kansiossa .tex-tiedoston kanssa.

# Kirjallisuusluettelon luominen

## Harjoitus 4.8

Edellä loit BibTeX-tiedostot Kemper.bib, Rudin.bib ja Vaisala.bib. Tuo teokset kirjallisuusluetteloon koodilla

```
\bibliographystyle{plain}  
\bibliography{Kemper,Rudin,Vaisala}
```

Sijoita koodi työsi loppuosaan, esimerkiksi juuri ennen komentoa `\end{document}`.

## Harjoitus 4.9

Viittaa kuhunkin teokseen työssäsi komennolla `\cite{viittaustunnus}`. Muista, että viittaustunnus poikkeaa useimmiten tiedoston nimestä.

# Käytäntö

Jotta viitteet tulisivat näkyviin, toimi seuraavasti:

- ▶ aja tiedosto PDFLaTeXilla
- ▶ aja tiedosto BibTeXilla (ja tarkista alapalkkiin ilmestyvät viestit!)
- ▶ aja tiedosto PDFLaTeXilla kahdesti

Jos viittaukset eivät tule näkyviin etkä tiedä mistä se johtuu, tarkista seuraavat kohdat:

- ▶ oletko käyttänyt viitatessa oikeita tunnuksia?
- ▶ oletko tuonut tiedostot  $\text{\LaTeX}$ iin oikein ja oikeilla nimillä?
- ▶ ovatko tiedostot todella muotoa `.bib`?
- ▶ ovatko tiedostot oikeassa kansiossa?

# Viittaustyyli

Viittaukset tulevat näkyviin hakasuluissa olevina numeroina. Esimerkiksi lähdeteoksen kirjoittajaa varten lukija joutuu siis tarkistamaan lähdeluettelon. Tämä on tyypillistä varsinkin matemaattisessa kirjallisuudessa.

Toisenlaista viittaustekniikkaa varten voi käyttää esimerkiksi natbib-järjestelmää, jota ei kuitenkaan tällä kurssilla käsitellä. Lisätietoa saa vaikkapa osoitteesta

[http://en.wikibooks.org/wiki/LaTeX/Bibliography\\_Management#Natbib](http://en.wikibooks.org/wiki/LaTeX/Bibliography_Management#Natbib)

## Sisällysluettelo ja kansilehti

Sisällysluettelon saa haluamaansa kohtaan työtä komennolla `\tableofcontents`. Tiedoston joutuu yleensä ajamaan useaan kertaan, ennen kuin numeroinnit tulevat näkyviin oikein.

Kansilehti luodaan komennolla `\maketitle`. Tätä varten esittelyosaan lisätään komennot `\title{otsikko}`, `\author{nimi}` ja `\date{pvm}`.

Kansilehti tulostuu joissain luokissa (esim. `book` ja `report`) omaksi sivukseen, `article`-luokassa osaksi ensimmäistä sivua.

## Harjoitus 4.10

Luo työllesi kansilehti ja sisällysluettelo. Kokeile miltä työsi näyttää, jos vaihdat luokaksi report tai book. Millaisia eroja huomaat?



## Dokumenttiluokan asetukset

Tiedoston aloittavalle komennolle `\documentclass[]{}` voi antaa valinnaisena argumenttina eräitä dokumentin ulkoasua koskevia asetuksia:

- ▶ Kirjainkoon vaihtoehdot ovat 10pt, 11pt ja 12pt.
- ▶ Luonnostilan saa valinnalla `draft`. Tällöin kääntäminen on nopeaa ja liian pitkät rivit tulevat merkityiksi.
- ▶ Tekstin saa jaettua kahteen kolumniin valinnalla `twocolumn`.
- ▶ `article`-luokassa tulostus on oletusarvoisesti yksipuoleinen - tämän voi vaihtaa valinnalla `twoside` (vast. `oneside`).

Useammat määritteet erotetaan toisistaan pilkuilla, kuten tavallista.

Hieman lisätietoa löytyy esimerkiksi osoitteesta

<http://texblog.org/2013/02/13/latex-documentclass-options-illustrated>.