

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

31. maaliskuuta 2015

# Suunnitelma

## 1. kerta

Intro

Dokumentin rakenne

Syntaksi

Otsikot ja korostukset

Matemaattinen teksti

## 2. kerta

Matemaattinen teksti

Lauseympäristö

Sisäiset viittaukset

Listarakenteet

Omat komennot

## 3. kerta

Kuvat

Taulukot

Matriisit

## 4. kerta

Viittaaminen kuviin ja

taulukoihin

Kirjallisuusviitteet

Dokumentin viimeistely

## 1. kerta

### Intro

Dokumentin rakenne

Syntaksi

Otsikot ja korostukset

Matemaattinen teksti

# Mikä L<sup>A</sup>T<sub>E</sub>X on?

- ▶ Tekstin ladontajärjestelmä
- ▶ Matemaattisen tekstin tuottamista varten
- ▶ Kehitetty T<sub>E</sub>X-kielen päälle (Leslie Lamport)
- ▶ Laajalti käytössä ympäri maailman
- ▶ Merkittävästi erilainen kuin WYSIWYG-järjestelmät

## 1. kerta

Intro

**Dokumentin rakenne**

Syntaksi

Otsikot ja korostukset

Matemaattinen teksti

# Dokumentin rakenne ja luominen

$\LaTeX$ -dokumentit tuotetaan tällä kurssilla helppokäyttöisellä Texmaker-ohjelmalla.

Texmakeriin kirjoitetaan dokumentin koodi ja lopuksi tiedosto *käännetään* eli *ajetaan* varsinaiseksi (PDF- tai DVI-) tiedostoksi. Kääntämisen hoitaa  $\LaTeX$  käyttäjältä piilossa.

Tarvittavat ohjelmat omalle koneelleen löytää esimerkiksi Matematiikan laitoksen sivujen kautta osoitteesta <http://wiki.helsinki.fi/pages/viewpage.action?pageId=62428926>

Kaikkein helpoimmin kuitenkin pääsee alkuun käyttämällä jotain online-ympäristöä kuten esimerkiksi Overleaf (<https://www.overleaf.com/>). Tällöin mitään ohjelmia ei tarvitse asentaa eikä tiedostoja siirrellä koneiden välillä.

## Harjoitus 1.1

- ▶ Avaa Texmaker ja luo uusi tiedosto
- ▶ Tallenna se muodossa `nimi.tex` johonkin tätä kurssia varten luomaasi kansioon
- ▶ Kirjoita tiedostoon seuraavat rivit:

```
\documentclass{article}
\begin{document}
Huhuu!
\end{document}
```

- ▶ Aja tiedosto PDFLaTeX:illa
- ▶ Valitse View PDF

Lopputuloksena tulisi olla PDF-tiedosto, jonka yläreunassa lukee "Huhuu!".

# Dokumentin rakenne ja luominen

L<sup>A</sup>T<sub>E</sub>X-dokumentti koostuu kahdesta osasta: *esittelyosasta*, joka sisältää tarpeellisia asetuksia ja *sisällöstä* eli varsinaisesta dokumentista.

- ▶ Tiedosto ja samalla esittelyosa aloitetaan komennolla `\documentclass{...}`, jolla valitaan dokumenttiluokka (`article`)
- ▶ Esittelyosaan lisätään komentoja tarpeen mukaan
- ▶ Komento `\begin{document}` aloittaa itse dokumentin ja lopettaa esittelyosan
- ▶ Komento `\end{document}` lopettaa dokumentin eikä sen perään kirjoitettuja rivejä käännetä.
- ▶ Varsinainen työn sisältö kirjoitetaan siis komentojen `\begin{document}` ja `\end{document}` väliin.

(Vrt. edellinen harjoitus!)



# Dokumentin rakenne ja luominen

Esittelyosassa eli heti komennon `\documentclass{}` jälkeen valitaan käytettävät paketit ja asetukset. Ääkkösiä ja suomalaista tavutusta varten otetaan käyttöön tietyt `inputenc-`, `fontenc-` ja `babel-` paketit:

## Harjoitus 1.2

Tee dokumenttiisi seuraavat muutokset (älä siis luo uutta tiedostoa):

```
\documentclass[a4paper]{article}
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[finnish]{babel}
\usepackage{geometry}
\begin{document}
Öö häh? Herätys!
\end{document}
```

Onnistuuko tiedoston ajaminen? Toimivatko ääkköset?

## Lisätieto

- ▶ Käyttämällä `inputenc`-pakettia  $\text{\LaTeX}$  lukee `.tex`-tiedoston `utf8`-koodattuna, jolloin esim. ä voidaan kirjoittaa tiedostoon sellaisenaan eikä muodossa `\"a`. Tämä vaatii, että käyttämäsi editori tallentaa tiedoston `utf8`-koodattuna. Esim. `Texmaker` ja `Overleaf` tekevät niin oletuksena. Jos ääkköset eivät toimi oikein, niin ongelma on todennäköisesti siinä, että tiedosto on koodattu eri tavalla kuin sitä yritetään lukea. Jos ääkköset eivät toimi, voit yrittää kirjoittaa `[utf8]` sijaan esim. `[latin1]`, `[ansinew]` tai `[applemac]`. Paras olisi kuitenkin käyttää `[utf8]:aa` ja yrittää saada editori tallentamaan tässä muodossa.
- ▶ Paketti `fontenc` ottaa käyttöön laajemmat fontit, jolloin esim. ä tulostetaan yhtenä merkkinä eikä niin, että merkit `a` ja `¨` kirjoitetaan päällekkäin. Tällöin vaikkapa tavutus ja pdf-haku saadaan toimimaan oikein.
- ▶ Lopuksi `babel` asettaa dokumentin kielen suomeksi ottamalla käyttöön suomalaiset tavutussäännöt ym.

## Harjoitus 1.3

Kopioi työhösi pari sivullista suomenkielistä tekstiä esimerkiksi Wikipediasta (vältä erikoismerkkejä) täytetekstiksi. Aja tiedosto. Varmista vielä ääkkösten ja tavutuksen toimiminen.

Edellä mainitut paketit ovat esimerkkejä *makropaketeista*, joilla  $\text{\LaTeX}$ in eli latojan toimintaan voi vaikuttaa. Matemaattista tekstiä varten on vielä syytä ottaa käyttöön muutama lisäpaketti.

## Harjoitus 1.4

Ota käyttöön paketit<sup>1</sup>

`amsmath`, `amsthm` ja `amssymb`

lisäämällä dokumenttisi esittelyosaan seuraavat komennot:

```
\usepackage{amsmath}  
\usepackage{amsthm}  
\usepackage{amssymb}
```

Nämä sisältävät fontteja, symboleita ja muuta matemaattisen tekstin kirjoittamiselle tarpeellista. Paketti `amsmath` on ladattava ennen `amsthm`-pakettia.

---

<sup>1</sup>ams tulee sanoista American Mathematical Society

## 1. kerta

Intro

Dokumentin rakenne

**Syntaksi**

Otsikot ja korostukset

Matemaattinen teksti

# Syntaksi eli "kielioppi"

L<sup>A</sup>T<sub>E</sub>Xin toimintaa ohjataan komennoilla. Niillä tuotetaan esimerkiksi matemaattisia symboleita, korostetaan tekstin osia, luodaan otsikoita, piirretään kuvia, määritellään asetuksia jne.

Komennot alkavat aina kenoviivalla \

Komentoja voi tarvittaessa etsiä esimerkiksi oppaista

<http://www.ntg.nl/doc/hellgren/lyhyt2e.pdf> ,

<http://en.wikibooks.org/wiki/LaTeX> ja

<http://www.rri.res.in/~sanjib/latex/ltx-2.html>

Komennot tarvitsevat usein *argumentin* (lisämääreen). Se kirjoitetaan komennon perään

- ▶ aaltosulkuihin {}, kun argumentti on pakollinen
- ▶ hakasulkuihin [], kun argumentti on valinnainen

Komennolla voi olla yksi tai useampi pakollinen argumentti ja lisäksi yksi tai useampi valinnainen argumentti. Toisilla komennoilla ei ole ainuttakaan argumenttia.

## Esimerkiksi

- ▶ Komennolla `\cup` ja `\cap` ei ole yhtään argumenttia (joukkojen yhdiste ja leikkaus)
- ▶ Komennolla `\sqrt [] {}` on yksi valinnainen ja yksi pakollinen argumentti (valinnainen on juuren kertaluku, pakollinen juurettava luku. Jos valinnainen argumentti puuttuu,  $\text{\LaTeX}$  tulkitsee neliöjuureksi)
- ▶ Komennolla `\documentclass [] {}` on myös valinnainen argumentti, jolla voi valita mm. kirjain- ja paperikoon
- ▶ Komennolla `\frac {} {}` on kaksi pakollista argumenttia (murtoluvun osoittaja ja nimittäjä)

Argumenttien järjestyksen ja määrän kanssa on oltava tarkka!



Muotoa

```
\begin{ympäristön nimi}  
  ...  
\end{ympäristön nimi}
```

olevilla komentopareilla käytetään ns. *ympäristöjä*. Näitä voivat olla esimerkiksi lauseet, määritelmät, listat ja taulukot.

Ympäristölläkin voi olla nimen lisäksi muita pakollisia tai valinnaisia argumentteja, kuten tieto taulukon sarakkeiden lukumäärästä tai kuvan toivotusta sijainnista.

Kurssin aikana opetellaan käyttämään muutamia tarpeellisia ympäristöjä ja luomaan omia komentoja.

# Erikoismerkeistä

Jotkin erikoismerkit on varattu  $\text{\LaTeX}$ in käyttöön:

- ▶ % aloittaa kommenttirivin
- ▶ \$ aloittaa ja lopettaa tavallisen matematiikkatilan
- ▶ \ aloittaa komennon
- ▶ & on käytössä kun rivejä järjestetään kohdakkain
- ▶ { ja } ovat komennon argumentin ympärille tulevat merkit

Jotta erikoismerkin saisi näkyviin lopullisessa työssä, on käytettävä komentoa:

Komento	Tulostus
<code>\%</code>	<code>%</code>
<code>\\$</code>	<code>\$</code>
<code>\textbackslash</code>	<code>\</code>
<code>\&amp;</code>	<code>&amp;</code>
<code>\{\}</code>	<code>{}</code>

## Harjoitus 1.5

Kolme peräkkäistä pistettä saa komennolla `\dots`. Kokeile, miten lopputulokset eroavat, jos kirjoitat pisteet itse. Kirjoita sitten seuraava:

Erikoismerkkejä ovat mm. %, \$ ja &. Merkkijonon `\textbackslash` tuottaminen onnistuu näin...

## Harjoitus 1.6

Kommenttirivin avulla koodin sekaan voi kirjoittaa selkeyttäviä huomautuksia, jotka eivät tule näkyviin lopulliseen työhön. Kommenttirivi aloitetaan merkillä % ja päätetään rivinvaihtoon.

Kirjoita koodin sekaan esimerkiksi rivi

```
% Tämä rivi ei tule näkyviin lopullisessa työssä.
```

ja testaa, pitääkö väite paikkansa.

# Virheilmoitukset

Tiedoston ajamisen jälkeen Texmakerin alareunaan saattaa ilmestyä sinisiä ja punaisia viestejä.

- ▶ **Siniset** viestit ovat varoituksia ulkonäöllisistä seikoista
- ▶ **Punaiset** viestit ovat kääntämisen estäviä virheitä

Punaisten virheilmoitusten kanssa on pakko olla tarkka – virheet tulee korjata heti!

Virheilmoituksesta on pääteltävissä jotain sattuneesta virheestä, mutta tämä vaatii totuttelua. Yleensä kyseessä on jonkin komennon väärinkirjoitus, puuttuva aaltosulku tai muu pieni yksityiskohta.

Virheilmoituksia saattaa tulla valtavasti vaikka kyse olisi yksittäisestä ongelmasta!

## 1. kerta

Intro

Dokumentin rakenne

Syntaksi

Otsikot ja korostukset

Matemaattinen teksti

# Otsikot

Teksti jäsennetään ja otsikoidaan komennoilla

- ▶ `\part{Osan otsikko}`
- ▶ `\chapter{Luvun otsikko}` (vain luokissa book ja report)
- ▶ `\section{Osion otsikko}`
- ▶ `\subsection{Aliosion otsikko}`
- ▶ `\subsubsection{Alialiosion otsikko}`
- ▶ `\paragraph{Kappaleen otsikko}`
- ▶ `\subparagraph{Alikappaleen otsikko}`

Neljän karkeimman tason otsikot numeroidaan automaattisesti. Numeroinnin saa pois lisäämällä merkin \* komennon perään, siis esimerkiksi

```
\section*{numeroimaton otsikko}
```

## Harjoitus 1.7

- ▶ Jaa tekstisi neljään numeroituun osioon
- ▶ Jaa ensimmäinen osio lisäksi muutamaksi ali- ja alialiosiksi
- ▶ Jätä ainakin yksi alialiosio numeroimatta
- ▶ Nimeä kaikki osiot

Tee ensimmäisen kerran harjoitukset ensimmäiseen osioon, toisen kerran harjoitukset toiseen osioon jne.

# Tekstin muokkaaminen

Tekstieditorissa eli Texmaker-ohjelmassa käytetty fontti tai tekstin suuruus eivät vaikuta lopulliseen työhön, vaan kaikki ulkoasulliset muutokset on tehtävä komennoilla.

Eryteisesti kannattaa huomata, että tekstiä kirjoittaessa

- ▶ peräkkäiset välilyönnit tulostuvat yhdeksi välilyönniksi
- ▶ myös yksittäinen rivinvaihto tulkitaan välilyönniksi
- ▶ tyhjä rivi (yksi tai useampi) aloittaa uuden kappaleen

Erikokoisia välilyöntejä varten on omia komentojaan, kuten esimerkiksi `\, , \quad` ja `\qquad`.

Yhden tyhjän rivin saa komennolla `\vspace{\baselineskip}`.



# Tekstin muokkaaminen

Tekstä voi *korostaa* komennolla `\emph{korostettava teksti}`, joka muuttaa fontin sopivaksi (kursiiviksi, pystyfontiksi tms.) kontekstin perusteella.

Tekstin **lihavointi** onnistuu komennolla `\textbf{lihavoitava teksti}` ja *kursivointi* komennolla `\textit{kursivoitava teksti}`.

Myös alleviivaus on mahdollista. Siihen kannattaa käyttää esim. `soulutf8`-paketin `\ul`-komentoa, joka sallii alleviivattujen sanojen tavutuksen.

## Harjoitus 1.8

Valitse dokumentistasi muutama rivi tekstiä ja muokkaa sitä käyttämällä korostuksia, kursivointia ja vahvennusta. Esimerkiksi siis jotain seuraavanlaista:

Kokeillaan *tekstin korostamista*. Sitten kursiivia: *kursivoitu teksti korostetaan* palauttamalla *fontti takaisin pystyyn*. Kuinkahan **lihavoitu teksti** korostuu?

Tässä sana "palauttamalla" on korostettu komennolla `\emph`, joka on komennon `\textit` argumentin sisällä:

```
\textit{... \emph{...} ...}
```

## 1. kerta

Intro

Dokumentin rakenne

Syntaksi

Otsikot ja korostukset

**Matemaattinen teksti**

# Matematiikkatila

Matemaattisia ilmaisuja saa tekstin sekaan käyttämällä komentoja `\(` ja `\)`. Esimerkiksi rivi

Yhtälöt `\(x^2+y^2-2=0\)` ja `\(y=2x+1\)` toteutuvat samanaikaisesti tasan kahdessa tason pisteessä.

tulostuu riviksi

Yhtälöt  $x^2 + y^2 - 2 = 0$  ja  $y = 2x + 1$  toteutuvat samanaikaisesti tasan kahdessa tason pisteessä.

Huomaa, että matematiikkatilassa kirjaimet tulostuvat erilailla, kuin muussa tekstissä.

## Harjoitus 1.9

Tuota seuraava lause dokumenttiisi:

Vaihdannaisessa renkaassa pätee  $(a + b)^2 = a^2 + 2ab + b^2$ . Jos siis  $2ab \neq 0$ , niin  $(a + b)^2 \neq a^2 + b^2$ .

Erisuuruuden saat komennolla `\neq`. Eksponentti kirjoitetaan tyyliin `x^2` (tulostaa  $x^2$ ).

# Kaavarivi

Kun matemaattinen ilmaisu halutaan yksinkertaisesti omalle kaavarivilleen, se kirjoitetaan komentojen `\[ ja \]` väliin. Esimerkiksi

Toisen asteen yhtälön `\(ax^2 + bx + c = 0\)` ratkaisu saadaan kaavasta

```
\[  
  x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.  
\]
```

tuottaa seuraavanlaisen esityksen:

Toisen asteen yhtälön  $ax^2 + bx + c = 0$  ratkaisu saadaan kaavasta

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.$$

## Harjoitus 1.10

Tuota seuraava dokumenttiisi:

Funktion  $f: \mathbb{R} \rightarrow \mathbb{R}$  derivaatta pisteessä  $x_0 \in \mathbb{R}$  on

$$f'(x_0) = \lim_{x \rightarrow x_0} \frac{f(x) - f(x_0)}{x - x_0},$$

mikäli raja-arvo on olemassa.

- ▶ Kaksoispisteen paikalla kannattaa käyttää komentoa `\colon`
- ▶ Symbolit  $\mathbb{R}$ ,  $\rightarrow$ ,  $\in$  ja  $\lim$  saa komennoilla `\mathbb{R}`, `\to`, `\in` ja `\lim`.
- ▶ Derivointipilkku tulee heittomerkillä `'`.
- ▶ Alaindeksin (symbolille tai operaattorille) saa kirjoittamalla `symboli_{alaindeksi}`.

## Harjoitus 1.11

Tuota seuraava dokumenttiisi:

Jos  $F$  on  $\sigma$ -algebra ja  $A_i \in F$  kaikilla  $i = 1, 2, \dots$ , niin

$$\bigcup_{i=1}^{\infty} A_i \in F.$$

Tarvitset mm. komentoja `\sigma`, `\in`, `\bigcup` ja `\infty`. Ala- ja yläindeksit kirjoitetaan merkkien `_` ja `^` avulla (jos indeksiin halutaan enemmän kuin yksi merkki, se täytyy laittaa aaltosulkeisiin kuten edellisessä tehtävässä).



Yleisimpiä matemaattisia symboleita löytää Texmakerin vasempaan laitaan avautuvista valikoista. Muuten niitä voi etsiä esim. seuraavista osoitteista:

- ▶ <http://www.tex.ac.uk/tex-archive/info/symbols/comprehensive/symbols-a4.pdf>
- ▶ <http://detexify.kirelabs.org/classify.html>

## Harjoitus 1.12

Selvitä, miten voit tuottaa vektorimerkinnät  $\bar{v}$ ,  $\bar{w}$  ja  $\overline{AB}$ . Kirjoita seuraava:

Vektoreiden  $\bar{v} = \overline{AB}$  ja  $\bar{w} = \overline{CD}$  ristitulo  $\bar{v} \times \bar{w}$  on kohtisuorassa kumpaakin vektoria vastaan. Vektoreiden pistetulo  $\bar{v} \cdot \bar{w}$  on sen sijaan reaaliluku.

## 2. kerta

Matemaattinen teksti

Lauseympäristö

Sisäiset viittaukset

Listarakenteet

Omat komennot

# Sulut

Varsinkin kaavariville kirjoitettaessa monet symbolit ovat huomattavan kookkaita. Sulut tulostuvat automaattisesti oikean kokoisina kun niille käytetään komentoja pelkkien merkkien ( ja ) sijaan.

Komento	Tulostus	Komento	Tulostus
<code>\left(</code>	(	<code>\right)</code>	)
<code>\left[</code>	[	<code>\right]</code>	]
<code>\left\lbrace</code>	{	<code>\right\rbrace</code>	}
<code>\left\langle</code>	<	<code>\right\rangle</code>	>
<code>\left </code>		<code>\right </code>	

Jokaista `\left`-alkuista komentoa täytyy seurata `\right`-alkuinen komento, vähintään `\right.`, joka ei tulosta mitään. Samoin `\right`-alkuista komentoa on edellettävä `\left`-alkuinen komento, vähintään `\left.`.

## Harjoitus 2.1

Tuota seuraava dokumenttiisi:

Todista seuraava yhtälö:

$$\{2^n \mid n \in \mathbb{Z}\} = \left\{ \left(\frac{1}{2}\right)^{-n} \mid n \in \mathbb{Z} \right\}$$

Komennolla `\middle|` saat pystyviivan automaattisesti oikean kokoisena ja komennolla `\,` hieman ylimääräistä väliä niiden ympärille. Tehtävästä **1.10** saat apua symboleiden  $\in$  ja  $\mathbb{Z}$  luomiseen.

## Harjoitus 2.2

Kirjoita seuraava:

Jos  $F$  on  $\sigma$ -algebra ja  $P: F \rightarrow \mathbb{R}$  todennäköisyys, niin tapahtumille  $A_1, A_2, \dots \in F$  pätee

$$P\left(\bigcup_{i=1}^{\infty} A_i\right) \leq \sum_{i=1}^{\infty} P(A_i).$$

# Vanhoja tapoja

- ▶ Vanha  $\text{T}_{\text{E}}\text{X}$ -tapa kirjoittaa kaava tekstin sekaan on  $\$kaava\$$ :  
Yhtälön  $\$2x+1=0\$$  ratkaisu on  $\$x=-1/2\$$ .
- ▶ Käännöksissä tuleva virheilmoitus "Missing \$ inserted." tulee tästä – se tarkoittaa, että yritit kirjoittaa joko
  - ▶ matematiikkaa kuten  $\backslash\text{frac}\{\}\{\}$  olematta matematiikkatilassa tai
  - ▶ ei-matematiikkaa kuten  $\backslash\text{section}\{\}$  matematiikkatilassa.
- ▶  $\text{T}_{\text{E}}\text{X}$  yritti itse väliaikaisesti korjata virheen lisäämällä dollarin jonnekin.
- ▶ Käännös saattaa tämän paikkauksen takia mennä läpi, mutta kyseessä on silti virhe, joka on korjattava!
- ▶  $\text{T}_{\text{E}}\text{X}$ -tapa kaavarivin tuottamiseen on  $\$\$kaava\$\$$ . Sitä ei pitäisi  $\text{\LaTeX}$ issa käyttää ollenkaan, vaikka jotkut niin tekevätkin...

# Pitkät kaavat

Toisinaan matemaattiset ilmaukset ovat niin pitkiä, etteivät ne mahdu yhdelle riville. Tällöin voidaan käyttää esimerkiksi ympäristöä `align*`, jolla rivit saadaan allekkain seuraavasti:

```
\begin{align*}
```

```
xyz &= abc\\
```

```
    &= bca\\
```

```
w &= cab
```

```
\end{align*}
```

$$xyz = abc$$
$$= bca$$
$$w = cab$$

Huomaa, ettei `align*`-ympäristöä tarvitse erikseen sijoittaa matematiikkatilaan.

## Pitkät kaavat

Ympäristölle `align*` rivin katkaisukohta kerrotaan komennolla `\\` ja tasauskohta merkillä `&`. Katkaisukohta täytyy löytyä kaikilta paitsi viimeiseltä riviltä. Sen sijaan tasauskohdan täytyy löytyä jokaiselta riviltä!

Pitkille kaavoille voi vaihtoehtoisesti käyttää ympäristöä `multiline*`, jolle kerrotaan vain rivien katkaisukohdat. Se asettelee rivit automaattisesti (yleensä vähän epämääräisesti).

Myös tähdettämiä ympäristöjä `align` ja `multiline` voi käyttää, jolloin jokainen rivi tulee numeroiduksi.



## Harjoitus 2.3

Kirjoita muutaman yhtälön ketju ja sijoita yhtälöt allekkain, tasaten haluamastasi kohdasta. Voit esimerkiksi derivoida vaiheittain funktion  $x^3 \sin(\cos(x))$  tai keksiä jotkin muut yhtälöt. Yhtälöiden ei tarvitse olla tosia. Komennolla `\sin` saa sinifunktion tulostettua pystyfontilla.

## Harjoitus 2.4

Kopioi osa edellisen tehtävän yhtälöketjua ja kirjoita se numeroituun tai numeroimattomaan `multline`-ympäristöön.

# Numeroidut kaavat

Komennolla `\begin{equation}...\end{equation}` saa luotua samanlaisen kaavarivin kuin komennolla `\[...\]`, mutta numeroinnilla varustettuna. Esimerkiksi

Einsteinin yhtälöön

```
\begin{equation}
```

$$E=mc^2$$

```
\end{equation}
```

viitataan myöhemmin.

tuottaa seuraavan:

Einsteinin yhtälöön

$$E = mc^2 \tag{1}$$

viitataan myöhemmin.

Numeroituihin kaavoihin viittaamiseen palataan tuonnempana.

## Harjoitus 2.5

Tuota seuraava dokumenttiisi:

Eksponttifunktiolla  $e^x$  on sarjakehitelmä

$$e^x = \sum_{n=1}^{\infty} \frac{x^n}{n!}. \quad (1)$$

Summamerkinän saat komennolla `\sum_{}`, osamäärän komennolla `\frac{ }{ }` ja äärettömän symbolin komennolla `\infty`.

## 2. kerta

Matemaattinen teksti

**Lauseympäristö**

Sisäiset viittaukset

Listarakenteet

Omat komennot

# Lauseympäristö

Paketti `amsthm` tarjoaa mahdollisuuden esittää mm. lauseet, lemmat ja määritelmät tyylikkäästi.

Tarvittavat ympäristöt määritellään esittelyosassa komennolla `\newtheorem{}{}`. Ensimmäisiin aaltosulkeisiin tulee nimi, jolla ympäristöä käytetään ja jälkimmäisiin ympäristön otsikko, joka halutaan näkyväksi lopullisessa työssä.

Komento `\newtheorem{esim}{Esimerkki}` loisi ympäristön, jota käytettäisiin komennolla `\begin{esim}...\end{esim}` ja jonka otsikko valmiissa työssä olisi Esimerkki.

## Harjoitus 2.6

Luo esittelyosassa ainakin ympäristöt Lause, Määritelmä ja Esimerkki. Käytä luomiasi ympäristöjä ainakin kerran.

## Harjoitus 2.7

Lauseiden todistuksia varten on oma ympäristönsä, jota käytetään komennoilla `\begin{proof}... \end{proof}`. Kirjoita edellisessä harjoituksessa luomallesi lauseelle jokin todistus. Todistukseksi kelpaa muutama rivi valitsemaasi tekstiä.

# Lauseympäristön tyyli

Lauseympäristön tyyli valitaan esittelyosassa komennolla `\theoremstyle{...}`. Valittavissa on tyylit `plain`, `definition` ja `remark`. Tyylin valinta vaikuttaa sitä seuraaviin komennolla `\newtheorem{}{}` luotuihin ympäristöihin.

Esimerkiksi kirjoittamalla

```
\theoremstyle{plain}
\newtheorem{lemma}{Lemma}
\newtheorem{lause}{Lause}

\theoremstyle{definition}
\newtheorem{maar}{Määritelmä}
```

ympäristöt `lemma` ja `lause` tulostuvat `plain`-tyylin mukaisesti ja ympäristö `maar` `definition`-tyylin mukaisesti.

## Harjoitus 2.8

Valitse harjoituksessa 2.6 luomillesi lauseympäristöille jotkin tyylit. Kokeile, miltä erilaiset tyylit näyttävät ja valitse mieleisesi!



# Lauseympäristöjen numerointi

Komennolla `\newtheorem{}{}` luotu ympäristö luo samalla uuden *laskurin*, jonka mukaan ympäristön toteutumat numeroidaan.

Laskuri voidaan myös asettaa toiselle laskurille *alisteiseksi* tai valita mielivaltaisesti. Erityisesti eri ympäristöt voivat käyttää samaa laskuria, jos niin halutaan.

Komennolla `\newtheorem{}{ } []` on valinnaisena argumenttina laskuri, jolle ympäristön numerointi halutaan alisteiseksi. Tämä voisi olla esim. laskuri `section`, jolloin ympäristö numeroitaisiin muodossa "osionNumero.ympäristönNumero".

# Lauseympäristöjen numerointi

Jos ympäristön halutaan käyttävän jotain tiettyä laskuria, käytetään komentoa `\newtheorem{}[laskurin nimi]{}`. Huomaa, että tässä valinnainen argumentti sijoittuu pakollisten väliin.

Esimerkiksi koodilla

```
\newtheorem{teor}{Teoreema}[section]
\newtheorem{lemma}[teor]{Lemma}
```

luodut Lemma- ja Teoreema-ympäristöt noudattavat samaa, laskurille `section` alisteista numerointia.

Kokonaan numeroimattoman lauseympäristön voi luoda komennolla `\newtheorem*{}{}`.

### Harjoitus 2.9

Aseta yksi lauseympäristöistäsi (esim. Lause) laskurille `section` alisteiseksi. Anna sitten toisen lauseympäristön (esim. Lemma) laskuriksi edellinen lauseympäristö.

### Harjoitus 2.10

Luo jokin numeroimaton lauseympäristö ja käytä sitä työssäsi.

## 2. kerta

Matemaattinen teksti

Lauseympäristö

**Sisäiset viittaukset**

Listarakenteet

Omat komennot

# Sisäiset viittaukset

$\LaTeX$ illa voi helposti viitata numeroituihin kohteisiin, kuten lauseisiin tai yhtälöihin. Viitattava kohde täytyy ensin nimetä komennolla `\label{nimi}` (nimi ei tulostu työhön). Tämän jälkeen viittaaminen onnistuu komennoilla

- ▶ `\ref{nimi}` (tulostaa viitattavan kohteen numeron)
- ▶ `\eqref{nimi}` (tulostaa kohteen numeron sulkujen sisällä)
- ▶ `\pageref{nimi}` (tulostaa sen sivun numeron, jolla kohde on)

Sisäisten viittausten kanssa tulee aina käyttää komentoja. Tällöin viittaukset pysyvät kohdallaan vaikka numeroinnit muuttuisivat työn edetessä.

# Esimerkki

Seuraavassa on osiolle "Asiaa" annettu tunnus `sec:asiaa`. Osioon viitataan muualla tekstissä komennolla `\ref`.

```
\section{Asiaa}\label{sec:asiaa}
Jotain asiaa\dots

% Muualla koodissa
Osiossa~\ref{sec:asiaa} puhuttiin asiaa.
```

Numeroidulle yhtälölle tai vaikkapa lauseelle tunnus annetaan ympäristön aloittavan komennon jälkeen:

```
\begin{equation}\label{eq:tunnus}
...
\end{equation}
```

Huom! Uuden viittauksen jälkeen työn joutuu ajamaan kahdesti, jotta numeroinnit tulevat näkyviin (kahden kysymysmerkin sijaan).

## Lisätieto

Jos  $\LaTeX$ -tiedoston nimi on `nimi.tex`, niin ensimmäisellä ajokerralla  $\LaTeX$  luo (tai päivittää) tiedoston `nimi.aux`, joka sisältää viittaustiedot (kokeile avata tiedosto). Toisella ajokerralla viittaustiedot luetaan tästä ja päivitetään lopulliseen tiedostoon. Tätä ei voi yhdellä ajokerralla tehdä, koska viittaukset voivat viitata myös tekstissä eteenpäin.

## Harjoitus 2.11

Viittaa harjoituksessa 2.5 kirjoittamaasi numeroituun yhtälöön. Käytä komentoja `\eqref{}` ja `\pageref{}`.

## 2. kerta

Matemaattinen teksti

Lauseympäristö

Sisäiset viittaukset

**Listarakenteet**

Omat komennot



# Listarakenteet

L<sup>A</sup>T<sub>E</sub>Xilla voi luoda listoja mm. ympäristöjen `itemize` ja `enumerate` avulla. Ensimmäinen on ranskalaiset viivat-tyyppinen, jälkimmäinen numeroi listan jäsenet. Näitä käytetään seuraavasti:

Tässä listani:

```
\begin{itemize}
  \item asia
  \item toinen asia
  \item kolmas asia
\end{itemize}
```

Tässä listani:

- asia
- toinen asia
- kolmas asia

Luettelointiin käytetyt symbolit voi tarvittaessa valita vapaasti.

Seuraavassa tehtävässä harjoitellaan sisäkkäisten listojen käyttöä.

## Harjoitus 2.12

Luo ainakin kolmen kohdan numeroitu lista haluamiasi asioista. Luo yhdeksi listan jäseneksi toinen lista ja yhdeksi tämän listan jäseneksi kolmas lista. Käytä (ainakin) viimeiseen listaan numeroimatonta `itemize`-ympäristöä.

## Harjoitus 2.13

Luo vielä yksi lista, mutta käytä tällä kertaa ympäristöä `description`. Se toimii kuten `itemize`, mutta pelkän komennon `\item` sijaan käytetään komentoa `\item[nimi]`, jossa nimi on vapaasti valitsemasi merkkijono.

## 2. kerta

Matemaattinen teksti

Lauseympäristö

Sisäiset viittaukset

Listarakenteet

**Omat komennot**

# Omat komennot

Omia komentoja luodaan esittelyosassa komennolla `\newcommand{}{}`. Ensimmäinen argumentti on komennon nimi, toiseksi argumentiksi tulee komennon sisältö. Esimerkiksi

```
\newcommand{\R}{\mathbb{R}}
```

luo komennon `\R`, joka tulostaa matematiikkatilassa symbolin  $\mathbb{R}$ . Komennon luomisen jälkeen kyseisen symbolin tuottaminen onnistuu helposti.

Omia komentoja luomalla voit yksinkertaistaa ja helpottaa omaa työtäsi. Jo parikin kertaa toistuva komentojen ketju kannattaa määritellä esittelyosassa yhdeksi yksinkertaiseksi komennoksi.

## Harjoitus 2.14

Luo komennot merkinnöille  $\mathbb{N}$ ,  $\mathbb{Z}$ ,  $\mathbb{Q}$ ,  $\mathbb{R}$  ja  $\mathbb{C}$ . Kirjoita seuraava:

Lukujoukot muodostavat tornin

$$\mathbb{N} \subset \mathbb{Z} \subset \mathbb{Q} \subset \mathbb{R} \subset \mathbb{C}.$$

Osajoukkorelaation saat komennolla `\subset`. Kokeile myös, mitä komennot `\subseteq`, `\subsetneq` ja `\supset` tuottavat. Miten saisit symbolit  $\supsetneq$  ja  $\supseteq$ ? Entä symbolin  $\not\subset$  (tähän tarvitset kaksi kommentoa)?

Kirjoita työhösi vielä seuraava:  $\mathbb{Z} \not\subset \mathbb{N}$ .

## Omat komennot

Komennolla `\newcommand{} [] {}` on valinnaisena argumenttina luotavan komennon argumenttien lukumäärä. Esimerkiksi

```
\newcommand{\set}[1]{ \left\lbrace #1 \right\rbrace }
```

loisi komennon `\set{}`, jolla voisi tuottaa joukkomerkin.

Yleensä

```
\(  
  \N = \set{0,1,2,3,\dots}  
\)
```

Yleensä  $\mathbb{N} = \{0, 1, 2, 3, \dots\}$

Komennon `\newcommand{} {}` käyttäminen aiheuttaa konfliktin, jos samanniminen komento on jo käytössä. Tällöin kannattaa nimetä oma komentonsa toisin.

## Harjoitus 2.15

L<sup>A</sup>T<sub>E</sub>Xissa ei ole valmista komentoa itseisarvofunktiota varten. Korjaa puute luomalla komento `\abs{}`, jonka argumentti on itseisarvomerkkien sisään tuleva lauseke. Tuota sen avulla seuraavat kolmioepäyhtälöt:

$$||x| - |y|| \leq |x + y| \leq |x| + |y|$$

ja

$$\left| \int_a^b f(x) dx \right| \leq \int_a^b |f(x)| dx.$$

Huomaa, että itseisarvomerkkien koon on syytä muuttua niiden sisältämän lausekkeen koon mukaan.

Määrätyn integraalin saat komennolla `\int_{ } ^{ }` ja pienen välin komennolla `\,` (ennen merkkiä  $dx$ ).

## 3. kerta

### Kuvat

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

### Taulukot

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

### Matriisit

- Matriisiympäristöt
- Lisätietoa



# 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 välettä.

Valinnaisella argumentilla voidaan skaalata, kiertää ja rajata liitettävää kuvaa.

# Kuvien liittäminen

Työkansioon tallennetun kuvan smile.jpg 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  $\LaTeX$ in päättää itse mihin kuva sijoitetaan eli tehdä kuvasta *kelluva*. Tämä onnistuu käyttämällä `figure-ympäristöä`.

# Kelluva kuva

```
\begin{figure}  
  \centering      % kuvan keskittämistä varten  
  \includegraphics[height=4cm]{smile}  
  \caption{Helppo hymyillä}  
\end{figure}
```



Kuva: Helppo hymyillä



## Figure-ympäristön sijainti

Kelluvien ympäristöjen sijoittaminen johonkin tiettyyn paikkaan voi olla haastavaa. Yleensä kannattaa antaa  $\text{\LaTeX}$ in sijoittaa ympäristö haluamaansa paikkaan ja viitata siihen tekstistä `\ref` ja `\pageref` komennoilla.

Kelluvan ympäristön sijoittaminen "tähän" (siihen, mihin sen koodin perusteella kuvittelisi tulevan) on kuitenkin mahdollista käyttämällä `float`-pakettia. Kun paketti on käytössä, `figure`-ympäristölle voi antaa valinnaisen argumentin `H` (*Here*), jolloin se ei lähde seilaamaan:

```
\begin{figure} [H]
  ...
\end{figure}
```

# Figure-ympäristön argumentit

## Harjoitus 3.2

Tuo työhösi pari kuvaa käyttäen `figure`-ympäristöä. Anna  $\LaTeX$ in päättää toisen sijoittamisesta ja sijoita toinen "tähän". Lisää vielä kuvateksti ainakin sille, jonka sijainnista ei ole tietoa.

## Lisätieto

Kuvan sijoitteluun voi vaikuttaa myös ilman `float`-pakettia antamalla valinnaiseksi argumentiksi osan tai kaikki merkeistä `!htbp`. Nämä kertovat, mihin sijoittaminen *sallitaan*. Merkillä `h` sijoittaminen *sallitaan* "tähän", merkillä `t` (jonkin) sivun yläreunaan, merkillä `b` sivun alareunaan ja merkillä `p` omalle sivulle. Merkki `!` tarkoittaa, että kauneudesta voi tarvittaessa tinkiä (mutta ei kovin paljoa).

Esimerkiksi `!hb` tarkoittaa, että kuvan voi sijoittaa "tähän" tai sivun alareunaan ja että rumuuttakin *sallitaan*. Argumentti `h` (tai `!h`) *ei* toimi samalla tavalla kuin `float`-paketin `H` syistä, jotka jätetään uteliaitten itse selvitettäväksi.

# 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. Runsaasti esimerkkejä (koodeineen) löytyy täältä:

<http://www.texample.net/tikz/examples/all/>

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.

## 3. kerta

### Kuvat

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

### Taulukot

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

### Matriisit

- Matriisiympäristöt
- Lisätietoa

# 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>	sarakkeenvaihto
<code>\\</code>	rivinvaihto
<code>\hline</code>	vaakaviiva
<code>\cline{i-j}</code>	vaakaviiva sarakkeiden i ja j välillä



# Tabular-ympäristö

Esimerkiksi koodi

```
\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 L<sup>A</sup>T<sub>E</sub>Xille.

## 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

## Harjoitus 3.7

Luo seuraava taulukko:

<b>Päiväpetolintuja</b>		
<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

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

<b>Päiväpetolintuja</b>			
<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 taulukon 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}[H]
  \centering
  \begin{tabular}{...}
    ...
  \end{tabular}
  \caption{Otsikko}
\end{table}
```

Ympäristön `table` sijoittelu noudattaa samoja sääntöjä kuin `figure`. Erityisesti `float`-paketin kanssa sille voi antaa valinnaisena argumenttina `H:n`, jolloin taulukko tulee "tähän".

## Harjoitus 3.9

Tee tehtävässä 3.8 luomastasi taulukosta kelluva ja anna sille jokin otsikko.

## 3. kerta

### Kuvat

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

### Taulukot

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

### Matriisit

- Matriisiympäristöt
- Lisätietoa

# 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:

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

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

- ▶ sarakkeet erotetaan toisistaan merkillä `&`
- ▶ 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:

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

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

## 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. Kaavan mukana automaattisesti venyvän nuolen saa komennolla `\xrightarrow{kaava}`.

## Lisätietoa matriiseista

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

Tämä on tarpeen, jos halutaan valita eri sarakkeisiin erilainen tasaus tai luoda pystyviiva matriisiin 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:

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

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

## 4. kerta

### Viittaaminen kuviin ja taulukoihin

#### Kirjallisuusviitteet

- Lainaukset ja alaviitteet

- BibTeX-järjestelmä

- Viittaustyyli

#### Dokumentin viimeistely

- Sisällysluettelo ja kansilehti

- Dokumenttiluokan asetukset

# Viittaaminen kuviin ja taulukoihin

Kelluviksi kirjoitetut kuvat ja taulukot tulevat numeroiduiksi (tarkemmin sanottuna niiden otsikot kuten kuvateksti), jolloin niihin viittaaminen onnistuu helposti komennoilla `\label{}` ja `\ref{}`.

## Harjoitus 4.1

Viittaa harjoituksessa 3.9 luomaasi kelluvaan taulukkoon. Taulukolle on ensin annettava tunnus komennolla `\label{}`, joka tulee otsikon luovan `\caption{}`-komennon jälkeen. Käytä viittaamiseen komentoja `\ref{tunnus}` ja `\pageref{tunnus}`.

## Harjoitus 4.2

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



## 4. kerta

Viittaaminen kuviin ja taulukoihin

Kirjallisuusviitteet

Lainaukset ja alaviitteet

BibTeX-järjestelmä

Viittaustyyli

Dokumentin viimeistely

Sisällysluettelo ja kansilehti

Dokumenttiluokan asetukset

# 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 `quote` ja `quotation`.
- ▶ Alaviitteen voi luoda komennolla `\footnote{teksti}`. Komento kirjoitetaan siihen kohtaan, johon alaviitteen merkki halutaan.

# Lukuvälit ja ajatusviiva

- ▶ Lukuväleissä oleva viiva kirjoitetaan --, esim.

Vertaa s.~39--42 ja s.~39-42.

tulostaa

Vertaa s. 39–42 ja s. 39-42.

- ▶ Tällaista viivaa käytetään suomen kielessä myös ajatusviivana:

`\LaTeX{}`ia ei opi ymmärtämään -- siihen vain tottuu.

tulostaa

$\LaTeX$ ia ei opi ymmärtämään – siihen vain tottuu.

### Harjoitus 4.3

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

### Harjoitus 4.4

Lisää työhösi alaviite komennolla `\footnote{}`.

# Kirjallisuusviitteet

$\LaTeX$ issa kirjallisuusviittaukset kannattaa hoitaa Bib $\TeX$ -järjestelmän avulla.

Tällöin lähdeteokset kirjataan erilliseen `.bib`-tiedostoon (tai -tiedostoihin).

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

Viittaaminen tapahtuu komennolla `\cite{tunnus}`, jossa tunnus on eräs `.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=8kx1j48DWM4C},  
  year={2010},  
  publisher={Springer Berlin Heidelberg}  
}
```

Tämä koodi kuvailee yhden teoksen, mutta yhdessä .bib-tiedostossa voi olla useammankin teoksen tiedot. Jokaiselle teokselle tulee vastaavanlainen "blokki".

## .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 lähde-teosta varten löytyy jokin sopiva luokka (ja tarpeen tullen sellaisen voi luoda itsekin).



## Harjoitus 4.5

Kopioi seuraava uuteen tiedostoon ja tallenna se nimellä  
lahteet.bib työkansioosi.

```
@book{kemper,  
  title={A Course in Commutative Algebra},  
  author={Kemper, G.},  
  isbn={9783642035456},  
  series={Graduate Texts in Mathematics},  
  year={2010},  
  publisher={Springer Berlin Heidelberg}  
}
```

# Kirjallisuusluettelon luominen

Kun `.bib`-tiedosto on tallennettu koneelle, voidaan luoda kirjallisuusluettelo. Kirjallisuusluetteloon tulee näkyviin ne teokset, joiden tiedot ovat olemassa ja joihin on viitattu ainakin kerran tai jotka eksplisiittisesti otetaan mukaan `\nocite`-komennolla.

Luettelo luodaan työn loppuosaan komennoilla

```
\bibliographystyle{luettelon tyyli}  
\bibliography{lista1,lista2,lista3,...}
```

Koodissa `lista $\alpha$`  ovat `.bib`-tiedostojen nimet ilman `.bib`-päättää. Huomaa myös, ettei pilkun jälkeen sallita välilyöntiä.

Edellisen koodin toimimisen edellytyksenä on, että tiedostot `lista1.bib` jne. ovat samassa kansiossa `.tex`-tiedoston kanssa.

# Kirjallisuusluettelon luominen

## Harjoitus 4.6

- ▶ Edellä loit tiedoston `lahteet.bib`. Luo kirjallisuusluettelo koodilla

```
\bibliographystyle{plain}  
\bibliography{lahteet}
```

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

- ▶ Käännä tiedosto. Mitään ei näytä tapahtuvan, mutta `.tex`-tiedostoasi vastaavaan `.aux`-tiedostoon on tullut uusia komentoja.
- ▶ Käännä nyt työsi BibTeXillä (se luo `.aux`-tiedoston pohjalta `.bbl`-tiedoston) ja sitten uudelleen PDFLaTeXilla.
- ▶ Saat varoituksen tyhjistä `thebibliography`-ympäristöstä. Se johtuu siitä, että lähdeviittauksia ei vielä ole. Lähdeluettelon otsikon pitäisi kuitenkin ilmestyä työhön.

# Kirjallisuusviitteen luominen

Jos .bib-tiedostossa on teos, jonka tunnus on kemper (kuten nyt on), niin siihen viitataan tekstissä seuraavaan tapaan:

```
Kirjassa \cite{kemper} sanottiin jotain algebrasta.
```

## Harjoitus 4.7

- ▶ Lisää työhösi viittaus toistaiseksi ainoaan lähteeseesi.
- ▶ Käännä. Viittaukseksi tulee vain [?], mutta .aux-tiedosto päivittyy.
- ▶ Aja BibTeX. Se päivittää .bbl-tiedoston uuden .aux-tiedoston pohjalta.
- ▶ Aja PDFLaTeX. Viittaus on edelleen [?], mutta lähde ilmestyi lähdeluetteloon.
- ▶ Aja PDFLaTeX vielä kerran, jolloin viittauskin tulee oikein.

## Harjoitus 4.8

Lisää tiedostoon lahteet.bib kirja, jonka nimi on Topologia I, kirjoittaja Jussi Väisälä, julkaisija Limes ry ja painovuosi 2000. Valitse viittaustunnukseksi topo1.

## Harjoitus 4.9

Lisää työhösi viittaus Väisälän kirjaan. Viittausten/lähteiden muuttamisen jälkeen työn kääntämisprosessi on siis

- ▶ PDFLaTeX
- ▶ BibTeX
- ▶ PDFLaTeX
- ▶ PDFLaTeX

## .bib-tiedostot Internetistä

Kirjojen teostietoja löytää BibT<sub>E</sub>X-muodossa varsin suurella todennäköisyydellä Google Books-palvelusta.

Kuhunkin teokseen liittyvän Tietoja teoksesta-sivun alalaidassa on kohta Vie sitaatti ja sen vieressä painike BibT<sub>E</sub>X, josta tiedoston voi ladata.

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

Käytännössä ainakin viittaustunnus on syytä vaihtaa helpommaksi.

## Harjoitus 4.10

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

## Harjoitus 4.11

- ▶ Lisää Rudin.bib lähdelistaan lisäämällä se `\bibliography`-komennon argumenttiin (ilman päätettä ja ei välilyöntejä).
- ▶ Kokeile kääntämistä (PDFLaTeX, BibTeX, 2xPDFLaTeX). Teos ei ilmesty lähdeluetteloon, koska siihen ei ole viittauksia.
- ▶ Lisää teos lähdeluetteloon `\nocite`-komennolla (toimii kuten `\cite`, mutta ei tulosta mitään).
- ▶ Käännä ja varmista, että teos tulee lähdeluetteloon.

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

- ▶ oletko kääntänyt työn oikein (latex, bibtex, latex, latex)?
- ▶ oletko käyttänyt viitatessa oikeita tunnuksia?
- ▶ oletko tuonut tiedostot  $\LaTeX$ iin oikein ja oikeilla nimillä?
- ▶ ovatko tiedostot todella muotoa `.bib`?
- ▶ ovatko tiedostot oikeassa kansiossa?



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)

## 4. kerta

Viittaaminen kuviin ja taulukoihin

Kirjallisuusviitteet

Lainaukset ja alaviitteet

BibTeX-järjestelmä

Viittaustyyli

Dokumentin viimeistely

Sisällysluettelo ja kansilehti

Dokumenttiluokan asetukset

# 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 haluttaessa `\date{pvm}`. Päivämäärän saa pois komennolla `\date{}`.

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

## Harjoitus 4.12

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

## Harjoitus 4.13

- ▶ Dokumenttia on helpompaa navigoida elektronisesti, kun viitteet toimivat linkkeinä. Ota tätä varten käyttöön paketti `hyperref`. Paketti kannattaa ladata kaikkien muiden pakettien jälkeen (kuitenkin ennen `\title` ym. komentoja).
- ▶ Kokeile kääntää dokumentti – linkkien ympärillä on nyt ruma laatikko. Paranna tilannetta antamalla `hyperref`-paketille argumentti `colorlinks=true` (ota mallia esim. `inputenc`-paketin lataavasta rivistä).
- ▶ Huom! `hyperref` ei oikein tykkää ääkkösistä esim. lauseympäristöjen nimissä ja viittaustunnuksissa. Voit muokata nämä ääkkösittömiksi tai antaa `hyperrefille` argumentin `unicode=true`, joka saattaa auttaa. Jos et saa `hyperrefiä` toimimaan, voit jättää tämän harjoituksen tekemättä.

# 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`).

Useimmat määritteet erotetaan toisistaan pilkuilla, kuten tavallista.

Hieman lisätietoa löytyy esimerkiksi osoitteesta

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