

TeX ja LaTeX

Matemaattisessa tekstissä esiintyy kaikenlaisia erikoismerkkejä, joiden tuottaminen useimmilla tekstinkäsittelyohjelmilla on hankalaa tai suorastaan mahdotonta. Kaavoja koskevat myös omat typografiset sääntönsä, joita ohjelmat eivät yleensä ota huomioon.

TeX on Donald E. Knuthin kehittämä ohjelma matemaattisen tekstin tuottamiseen. TeX ei ole wysiwyg-ohjelma (what you see is what you get), vaan pikemminkin jonkinlainen ohjelmointikieli. Tekstin joukkoon sijoitetaan muotoilukomentoja, joiden perusteella ohjelma tuottaa siitä tulostettavan tai katseltavan version.

LaTeX on alkuperäisen TeXin päälle rakennettu aliohjelmakirjaston (tai oikeammin makrokirjaston) tapainen järjestelmä. Sen käyttö on usein vaivattomampaa, koska monet asiat on määritelty valmiiksi. Toisaalta joidenkin oletusarvojen muuttaminen voi olla työlästä.

TeXin ja LaTeXin syöttötiedosto on aivan tavallista tekstiä, joka voidaan kirjoittaa millä tahansa tekstieditorilla. Tiedostossa ei ole mitään tulostumattomia erikoismerkkejä.

Tiedosto sisältää vain itse tekstin ja muotoilukomentoja (nekin tekstiä), joiden osuus on suhteellisen pieni varsinkin suorasanaaisessa tekstissä. Tiedostot ovat paljon pienempiä kuin vastaavan asian sisältävät Wordin doc-tiedostot.

TeX ja LaTeX ovat saatavina useimpiin järjestelmiin. Ne sisältyvät Linuxin asennuspaketteihin.

Matemaattisten kaavojen tuottaminen erittäin helppoa.

Oikea typografias lähes automaattista. Ohjelmat on kuitenkin laadittu ennen matemaattisia merkintätapoja koskevaa ISO-standardia, joten joitakin yksityiskohtia on koodattava käsin, jos haluaa todella oikeaoppista jälkeä.

Monet tieteelliset lehdet ja kirjasarjat käyttävät tekstiensä ladontaan LaTeXia. Niiden verkkosivujen kautta voi ladata valmiita paketteja, joiden avulla artikkelin tai kirjan saa juuri kustantamon haluamaan muotoon.

Nykyisin monet lehdet ottavat vastaan vain omien ohjeittensa mukaisesti laadittuja LaTeX-tiedostoja, joten ohjelman käytön opiskelu on oikeastaan välttämätöntä, jos aikoo julkaista tieteellisiä artikkeleita.

TeX

Esimerkki: Kirjoitetaan vaikka tiedosto `koe.tex`, jossa on seuraava teksti:

```
Fermat'n suuren lauseen mukaan yht\ "al\ "oll\ "a
$$
  x^n + y^n = z^n
$$
ei ole ratkaisuja, joissa $x$, $y$ ja $z$ ovat
kokonaislukuja, jos $n > 2$.

Fermat'n pieni lause puolestaan sanoo, ett\ "a jos
$p$ on alkuluku ja $a$ jokin kokonaisluku,
jakolaskusta $a^p/p$ j\ "a\ "a jakoj\ "a\ "ann\ "okseksi $a$.

\bye
```

Kenoviiva `\` kertoo, että sitä seuraava sana on jokin TeX in toimintaa ohjaava komento. Sitä käytetään myös joidenkin poikkeuksellisesti käsiteltävien merkkien kuten aksenttien yhteydessä.

Syötetään tämä TeX-ohjelmalle:

```
tex koe
```

Jos tiedosto oli oikein kirjoitettu, tuloksena on tiedosto `koe.dvi`. Nimen loppuosa tulee sanoista *device independent*. Kyseessä on laiteriippumaton kuvaus, joka sisältää tiedon tulostettavista merkeistä ja niiden paikoista.

Tiedostoa `koe.dvi` ei voi sellaisenaan katsella eikä tulostaa. Sen avulla voidaan kuitenkin tuottaa monille erilaisille laitteille sopivia tiedostoja. PostScript-tiedosto voidaan tehdä sanomalla

```
dvips koe
```

Tämä tekee vastaavan PS-tiedoston ja lähettää sen saman tien oletustulostimelle. Yleensä tätä ei haluta, vaan mieluummin katsotaan ensin kuvaruudulla, onko lopputulos halutunlainen. Siksi talletetaankin tulos tiedostoon esimerkiksi komennolla

```
dvips -o koe.ps koe
```

Dvips:n konfigurointitiedostossa voidaan myös määritellä, että tämä on oletuksena.

Dvips tuottaa tiedoston `koe.ps`, jota voidaan tarkastella X-ikkunointiympäristössä `ghostview`-ohjelmalla:

```
gv koe.ps
```

Dvi-tiedostosta voidaan myös tehdä PDF-tiedosto:

```
dvipdf koe
```

Jos mukana on kuvia, PDF-muotoinen tiedosto voi olla huomattavasti pienempi kuin PS-tiedosto. Syntyvää tiedostoa voidaan katsella ohjelmalla `xpdf`

```
xpdf koe.pdf
```

Jos tulos on sellainen, kuin halusimme, se voidaan lähettää tulostimelle:

```
lpr koe.ps
```

Tulostus voidaan tehdä myös ohjelman `xpdf` ikkunassa näkyvän tulostimen ikonin avulla. Tällä tavoin kuvia sisältävät tiedostot tulostuvat yleensä paljon nopeammin.

Paperille pitäisi nyt ilmaantua seuraavanlainen tulostus:

Fermat'n suuren lauseen mukaan yhtälöllä

$$x^n + y^n = z^n$$

ei ole ratkaisuja, joissa x , y ja z ovat kokonaislukuja, jos $n > 2$.

Fermat'n pieni lause puolestaan sanoo, että jos p on alkuluku ja a jokin kokonaisluku, jakolaskusta a^p/p jää jakojäännökseksi a .

Välilyöntien määrällä ei ole merkitystä.

Jokainen kappale muotoillaan kokonaisuutena niin, että sanavälit ovat suunnilleen yhtä pitkiä. Pisteiden jälkeen tulee automaattisesti pitempi väli.

Kirjainten välistykset ovat kiinteät; sanan kirjaimia ei harvenneta, kuten joissakin muissa ohjelmissa on oletuksena.

Tavutusta pyritään välttämään. Oletuksena englanninkielen mukainen tavutus. Tavutusvihjeen voi merkitä: `kaivos\ -aukko`.

Kappaleet erotetaan ainakin yhdellä tyhjällä rivillä. Tulostuksessa oletus on, että kappaleen alku sisennetään, mutta kappaleiden väliin ei jätetä väliä.

Kenoviiva `\` tarkoittaa jotakin TeXille tarkoitettua ohjaustietoa. Jos sitä seuraa jokin sana, kyseessä on muotoilua ohjaava komento, joka ei tulostu näkyviin. Muuten kyseessä on erikoista käsittelyä vaativa merkki.

Skandinaaviset kirjaimet ovat `\ "a` ja `\ "o`. Samalla menetelmällä kirjaimiin voidaan lisätä muitakin aksenttimerkkejä. Tämä on hankalaa kirjoitettaessa suomenkielistä tekstiä, mutta ongelma voidaan välttää määrittelemällä editorin komento, joka muuttaa tekstin skandimerkit sopivaan muotoon tai pienellä muunnosohjelmalla.

Dollarimerkkien välinen osa on ns. matematiikkamoodissa. Kirjain-symbolit oletetaan muuttujiksi, jotka kursivoidaan.

Kaksinkertaisten dollarimerkkien välinen osa on omalle rivilleen tuleva kaava.

Ryhmittelyyn käytetään aaltosulkuja $\{, \}$.

Merkki \wedge tarkoittaa, että seuraava merkki on yläindeksi tai eksponentti. Operaatio kohdistuu vain seuraavaan merkkiin. Jos eksponentti on pitempi, se on kirjoitettava aaltosulkuihin:

$$x^{\{2n\}}, 1.5 \times 10^{\{-12\}}$$

Alaviiva $_$ tarkoittaa, että seuraava merkki on alaindeksi.

Prosenttimerkki tarkoittaa, että rivin loppuosa tulkitaan kommentiksi, joka ei tulostu.

Seuraavilla merkeillä on erikoismerkitys:

$\{ \} \wedge _ \& \# \% \$$

Jos näitä tarvitaan tekstissä, ne on kirjoitettava muotoon

$\$\lbrace$ \rbrace $\wedge _ \& \# \% \$$$

Matematiikkamoodin ominaisuuksia

Kreikkalaiset kirjaimet saadaan kutsuilla α , β , γ jne. Isot kirjaimet esimerkiksi Γ , Δ .

Murtolausekkeet

$$\frac{1}{x} \quad \frac{a + bx}{1 + cx + dx^2}$$

$$\frac{1}{x} \quad \frac{a + bx}{1 + cx + dx^2}$$

Integraalimerkki on \int . Integroimisrajat voidaan ilmoittaa samalla tavoin kuin ala- ja yläindeksit.

Esimerkiksi

$$u = \int f(x) dx, \quad \int_0^{\infty} e^{-x} dx = 1.$$

tuottavat

$$u = \int f(x) dx, \quad \int_0^{\infty} e^{-x} dx = 1.$$

Summamerkkiä \sum käytetään vastaavalla tavalla

$$\zeta(x) = \sum_{i=1}^{\infty} \frac{1}{i^x}.$$

$$\zeta(x) = \sum_{i=1}^{\infty} \frac{1}{i^x}.$$

Alkeisfunktiot

$$\sqrt{1+x} \quad \sin x \quad \tan x \quad \ln x$$

$$\sqrt{1+x} \quad \sin x \quad \tan x \quad \ln x$$

Näistä voi rakentaa mutkikkaampia lausekkeita:

`\sqrt{ {1+x \over 1- \sqrt{1-x}} }`

$$\sqrt{\frac{1+x}{1-\sqrt{1-x}}}$$

Sekalaista

`\lim_{x \rightarrow 0} = \infty \quad \text{\vert \pm 1 \vert} = 1`

$$\lim_{x \rightarrow 0} = \infty \quad |\pm 1| = 1$$

Kaikenlaisia erikoismerkkejä löytyy melkoinen määrä:

`\cup, \cap, \subset \wedge, \vee,`
`\rightarrow, \uparrow, \dots`

$\cup, \cap, \subset \wedge, \vee, \rightarrow, \uparrow, \dots$

Lausekkeiden ympärillä pitäisi olla oikean kokoiset sulut. Sulkumerkkien koon voi valita itse, mutta se käy myös automaattisesti makroilla `left` ja `right`. Esimerkiksi

`\left({a \over \sqrt{1+x}} \right)^2`

tuottaa lausekkeen

$$\left(\frac{a}{\sqrt{1+x}}\right)^2$$

Makrot

Jokin rakenne voi toistua tekstissä monia kertoja. Olisi kätevää, jos sen voisi määritellä vain kerran jonkinlaisena aliohjelmanä, jota kutsutaan aina tarpeen tullen. Jos emme ole tyytyväisiä muotoiluun, tarvitsee vain muuttaa tuon aliohjelman määrittelyä. Juuri näin voidaan tehdä \TeX issä, mikä erottaa sen useimmista muista tekstinkäsittelyohjelmasta.

Ohjelmointikielten terminologian mukaan \TeX issä määriteltävät oliot eivät ole aliohjelmaa vaan makroja.

Yksinkertaisimmillaan makro voi vain korvata tietyn pätkän tekstiä.

```
\def\name{What is the name of this book?}
```

Tässä `name` on makron nimi. Kun tekstissä esiintyy makrokutsu `\name`, se korvautuu tekstillä `What is the name of this book?`

Makrolla voi olla myös parametreja. Parametrien merkkeinä \TeX issä ovat `#1`, `#2` jne.

Ajatellaanpa, että tekstimme vilisee funktion f osittaisderivaattoja eri muuttujien suhteen. Määritellään seuraavanlainen makro:

```
\def\pd#1{{\partial f\over\partial #1}}
```

Nyt voimme kirjoittaa tiedostoon `pd{x}`, ja kas kummaa, siitä tulostuu

$$\frac{\partial f}{\partial x}$$

Makrojen avulla mutkikkaatkin rakenteet voidaan korvata yksinkertaisella kutsulla, jolle voidaan antaa aiheeseen sopiva havainnollinen nimi.

LaTeX

LaTeX on T_EXin päälle rakennettu makrokirjasto, jossa on valmiina joitakin perusmäärittelyjä artikkeleita ja kirjoja varten. Jos oletusmäärittelyt kelpaavat, käyttäjä pääsee itse asiaan hyvin vähällä vaivalla. Tilanne on sama myös, jos lähettää artikkelin tieteelliseen aikakauslehteen käyttämällä sen omia määrittelyjä.

Edellä ollut esimerkki LaTeXilla olisi:

```
\documentclass[a4]{article}
\begin{document}
\usepackage{amsmath}

Fermat'n suuren lauseen mukaan yht\ "al\ "oll\ "a
\begin{equation*}
x^n + y^n = z^n
\end{equation*}
ei ole ratkaisuja, joissa  $x$ ,  $y$  ja  $z$  ovat
kokonaislukuja, jos  $n > 2$ .

Fermat'n pieni lause puolestaan sanoo, ett\ "a jos
 $p$  on alkuluku ja  $a$  jokin kokonaisluku,
jakolaskusta  $a^p/p$  j\ "a\ "a jakoj\ "a\ "ann\ "okseksi  $a$ .

\end{document}
```

Tässä määritellään, että käytetään A4-kokoisia sivuja ja artikkelille määriteltyä ulkoasua.

Jos esimerkki on taaskin tiedostossa `koe.tex`, siitä saadaan dvi-tiedosto komennolla

```
latex koe.tex
```

Tästä eteenpäin toimintaa jatketaan samalla tavoin kuin T_EX-tiedoston tapauksessa.

LaTeXissa on valmiina työkalut sisällysluettelon ja aakkosellisen hakemiston tuottamiseen. Tiedot tulostuvat omiin tiedostoihinsa, joista ne poimitaan tekstin joukkoon LaTeXin komennoilla.

Kaavoille, kuville ja taulukoille voidaan antaa nimet, jolloin niihin voidaan viitata näiden nimien avulla. Tämä on kätevää, koska kaavojen ja kuvien numerot tuppaavat muuttumaan tekstiä muokattaessa. Koska viittaukset voivat osoittaa myös olioihin, jotka esiintyvät tekstissä vasta myöhemmin, niitä ei voida käsitellä saman tien. Nimet talletetaan tiedostoon, josta ne luetaan seuraavalla kerralla.

Jos tekstiä muutetaan, hakemiston sivunumerot ja muut viittaukset voivat muuttua, jolloin aputiedostot eivät ole ajan tasalla. Kun teksti on ajettu LaTeXin läpi ilman virheilmoituksia, LaTeX on suoritettava vielä uudestaan, jotta kaikki viittaukset olisivat oikein.

LaTeXin käyttö on aloittelijallekin suhteellisen helppoa, varsinkin jos voi käyttää jotakin valmiiksi määriteltyä muotoilutapaa. Jos sen sijaan haluaa luoda aivan oman tyylinsä, homma voi käydä aika työlääksi, koska jotkin asiat on haudattu syvälle makrokirjastoon, josta niiden löytäminen ja muuttaminen voi olla erittäin hankalaa.

Virhediagnostiikka on toinen pieni mutta harmittava ongelma. LaTeXin tapauksessa virhe saatetaan havaita jossakin syvällä sisäkkäisten makrokutsujen uumenissa, jolloin seurauksena voi joskus olla suuri joukko virheilmoituksia, jotka eivät ole mitenkään havainnollisia. TeXin ilmoitukset osoittavat selvemmin virheen todellisen paikan ja syyn.

AMS-LaTeXissa on matematiikkaan liittyviä laajennuksia. Ne saadaan käyttöön komennolla

```
\usepackage{amsmath}
```

Seuraavassa käsiteltävistä piirteistä jotkin ovat peräisin juuri tästä pakkauksesta.

Jollakin erikoistavalla käsiteltäviä ympäristöjä rajataan begin–end pareilla; esimerkiksi omalle rivilleen tuleva kaava on muotoa:

```
\begin{equation}
\cdots
\end{equation}
```

Näin kirjoitetulle kaavalle annetaan myös automaattisesti numero. Jos kaavaa ei haluta numeroida, kuten aikaisemmassa esimerkissä, sanan equation loppuun kirjoitetaan *.

Usealle riville tuleva kaava kohdistetaan aligned-ympäristössä. Kohdistuspaikka ilmoitetaan merkillä &.

Taulukkoja varten on tabular-ympäristö.

```

\documentclass{article}

\usepackage{amsmath}

\begin{document}

\title{Test file}
\author{H. Karttunen}

\maketitle

\section{Formula}

Using {\em Kepler's third law} we get
\begin{equation*}
\begin{aligned}
P^2 &= \frac{4\pi^2 a^3}{G(m_1+m_2)} \\
&= \frac{4\pi^2 \times 1}{6.67 \times 10^{-11} (5+5)} \\
&= 5.9 \times 10^{10} \text{ \rm s}^2.
\end{aligned}
\end{equation*}

\section{Table}

\begin{tabular}{|l|crr|} % preamble
\hline
name & type & magnitude & distance \\
\hline
M31 & Sb & 3.2 & 760 \\
M33 & Sc & 5.6 & 790 \\
\hline
\end{tabular}

\end{document}

```

Kuvat

Tekstiin voi liittää myös kuvia. Kuvien on oltava EPS-muotoisia (Encapsulated PostScript) tiedostoja. Nämä ovat PostScript-kielisiä tiedostoja, joiden täytyy toteuttaa joitakin rajoituksia. Idea on, että kuvat eivät aiheuta sivuvaikutuksia, jolloin niitä voi sijoitella melko vapaasti eri dokumentteihin. Valitettavasti monet piirto-ohjelmat vähät välittävät näistä vaatimuksista ja tuottavat sääntöjen vastaista koodia. Sellaisten kuvien vaikutukset voivat olla arvaamattomat. Pahimmassa tapauksessa ne sekoittavat tai jumittavat tulostimen.

EPS-tiedostot ovat itse asiassa ohjelmia ja sisältävät aivan tavallista tekstiä. Alussa on tavallisesti seuraavanlainen rivi:

```
%!PS-Adobe-2.0 EPSF-3.0
```

Numerot voivat poiketa tästä; ne ilmoittavat vain käytetyn kielen version. Varsinkin Windows-ympäristöstä peräisin olevissa tiedostoissa saattaa tämän edellä olla koodia, joka ei ole selväkielistä tekstiä. Tämä osa on poistettava ennen kuin kuvaa voi käyttää. Samoin tiedoston lopussa voi olla ylimääräistä roskaa. Tiedoston voi korjata millä tahansa editorilla.

Tiedoston alussa on joukko %-merkeillä alkavia kommenttirivejä, joilla on tietoa kuvasta. Niiden joukossa pitäisi olla seuraavantapainen rivi, joka määrittelee, minkä kokoisen tilan kuva tarvitsee. Yksikkönä on piste, joka on 1/72 tuumaa. Tätä tietoa tarvitaan kuvan sijoittelussa.

```
%%BoundingBox: 0 0 200 500
```

Jotta kuvia voitaisiin liittää LaTeX-dokumenttiin, tarvitaan joitakin apuvälineitä. Kuvien käsittelyyn tarvitaan jokin makropaketti, kuten `epsfig.sty`; muitakin on, mutta tämä lienee tavallisin. Paketti otetaan käyttöön `usepackage`-kutsulla:

```
\documentclass[a4]{article}
\usepackage{epsfig}
\begin{document}
...
```

Jos EPS-kuva on tiedostossa `kuva.eps`, se voidaan liittää dokumenttiin seuraavasti:

```
\begin{figure}
\epsfig{file=kuva.eps}
\caption{Tm on kuvateksti.}
\end{figure}
```

Kuvan `BoundingBox`in perusteella sille varataan sopiva tila. Kuvan kokoa tai asentoa voi myös muuttaa:

```
\epsfig{file=kuva.eps, scale=0.5} % pienennys puoleen
\epsfig{file=kuva.eps, width=10cm} % maksimileveys 10 cm
\epsfig{file=kuva.eps, height=10cm} % maksimikorkeus 10 cm
\epsfig{file=kuva.eps, angle=90} % kierto 90 astetta
```

Kuvaa ei sijoiteta siihen kohtaan tekstiä, jossa nämä kutsut esiintyvät, vaan seuraavan sivun alkuun. Jos samalle sivulle tulisi useita kuvia, ne eivät välttämättä mahdu sivulle, jolloin ne voivat siirtyä myöhemmille sivuille. Mikäli kuvia on paljon, viimeiset kuvat saattavat tulostua vasta koko tekstin jälkeen.

Kuva voidaan myös pakottaa tulostumaan juuri tähän parametrilla `H` (here):

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

Kuvan sijoitteluun liittyy muitakin parametreja. Ne löytyvät `epsfigin` mukana tulevasta dokumentista.

Typografiasta

Seuraavassa on sekalainen valikoima typografiaa koskevia sääntöjä, joissa tehdään usein virheitä.

- Matemaattisten funktioiden ja muuttujien nimet ovat yleensä kursivilla. Matematiikkamoodissa kursivointi tapahtuu automaattisesti. Poikkeuksena ovat funktiot, joiden nimessä on useita kirjaimia, kuten \sin , \cos , \ln . Näille on omat merkintänsä `\sin`, `\cos`, `\ln`.
- Nykyisen standardin mukaan myös vakiot merkitään tavallisella ”pystyfontilla”. Tällaisia vakioita ovat esimerkiksi e ja i . Niille voi määritellä omat makronsa, esimerkiksi

```
\define\e{\rm e}
```

- Myös derivointioperaattori d pitää kirjoittaa pystyfontilla.
- Integraalissa esiintyvän differentiaalilin eteen pitää jättää pieni väli. Se käy näin:

```
\int f(x)\,{\rm d}x
```


Kirjasintyytit

Tietokoneen ruudulla tai laserkirjoittimen paperille tulostama sivu koostuu kuvaelementeistä eli pikseleistä. Yksinkertaisimmassa tapauksessa pikseli voi olla vain musta tai valkea. Kirjaimet ja muut merkit ovat tavallaan matriiseja, joiden alkiot kertovat, onko jokin pikseli musta vai valkea.

Pikselifontti kuvaa merkit tällaisina matriiseina. Jos merkkiä suurennetaan, se alkaa näyttää kasalta nelikulmaisia laatikoita. Jos merkit on määritelty vain pikselien avulla, niiden kokoa ei voi muuttaa ilman, että tulos alkaa näyttää rumalta. Siksi pikselifontit on määriteltävä erikseen jokaista tarvittavaa kirjainkokoja varten.

Ns. *outline-fonteissa* merkit kuvataan matemaattisten käyrien avulla. Tällainen fontti on vapaasti skaalattavissa mihin kokoon tahansa, sillä määrittelytapa on erotuskyvystä riippumaton. Tulostuslaitteen tehtäväksi jää merkkien rasterointi tulostimen erotuskykyä vastaavaksi. Tällaista menetelmää käytetään mm. Adoben kehittämässä PostScript-kielessä.

Valitettavasti $\text{T}_{\text{E}}\text{X}$ syntyi ennen PostScriptiä. Sen alkuperäiset kirjasintyytit ovat pikselifontteja, jotka on luotava kullekin tulostimelle ja kirjasinkoolle. Kirjasintyyppien käsittelyä varten Knuth kehitti sitten toisen ohjelman Metafontin. Siinä merkit määritellään PostScriptin tapaan matemaattisten käyrien rajaamina alueina, ja näiden määrittelymien avulla tuotetaan vastaavat pikselitiedostot.

Linuxin $\text{T}_{\text{E}}\text{X}$ -järjestelmä generoi tarvittavat pikselitiedostot Metafont-ohjelmalla. Käyttäjän ei tarvitse huolehtia siitä, onko sopivankokoisia merkkejä määritelty.

Kun `dvipstai` jokin muu laiteriippuvaa koodia tuottava ohjelma suoritetaan, se tarkistaa, ovatko tarvittavat fontit jo olemassa. Ellei, se tuottaa ne Metafontilla. Tästä aiheutuu suuri määrä tulostusta, mutta siitä ei tarvitse huolestua.

Oletuksena käytetään `cm-sarjan` (computer modern) fontteja. Ne ovat varsin luettavia, vaikkakin vaikuttavat ehkä hieman vanhanaikaisilta. Ne sopivat kuitenkin yhteen kaavoissa käytettävien fonttien kanssa. Mikäli haluaa vaihtaa johonkin kovin toisenlaiseen kirjaimistoon, voi joutua melkoiseen suohon symbolifonttien kanssa.

Kirjallisuutta

Donald E. Knuth: *The T_EXbook*, Addison-Wesley 1984. Tämä on T_EX-ohjelman perusteos. Paikoitellen oikein hyvä ja hauska oppikirja, paikoitellen varsin vaikeaselkoinen, joskus suorastaan hämäräpääinen. Silti tämä on välttämätön käsikirja sille, joka haluaa todella perehtyä T_EXiin. Teoksesta on otettu jo kymmeniä painoksia, ja sitä on edelleen saatavana.

Leslie Lamport: *LaTeX*, Addison-Wesley 1986. LaTeXin kehittäjän alkuperäinen käsikirja. LaTeXia on myöhemmin kehitetty edelleen, ja tämä teos ei sisällä näitä uudempia piirteitä.

Goossens, Mittelbach, Samarin: *The LaTeX companion*, Addison-Wesley 1994. Suhteellisen hyvä perusteos, joka käsittelee uudempaa standardia LaTeX2 ϵ . Tämä tai seuraava kirja ovat hyödyllisiä (mitteipä välttämättömiä) hankintoja LaTeXin käyttäjille.

Lipkin: *LaTeX for Linux*, Springer 1999. Samantapainen kuin edellinen. Keskittyy Linux-käyttöjärjestelmään, mutta TeX ja LaTeX ovat hyvin samanlaisia kaikissa järjestelmissä.