

Teemu Likonen

Käytännöllistä L^AT_EXia

Latex-ladontajärjestelmän opas
Versio 2022.5

Käytännöllistä Latexia

TEKIJÄ: Teemu Likonen <tlikonen@iki.fi>

VERSIO: 2022.5

PÄIVÄYS: 28.5.2022 kello 13:38 (ensijulkaisu: 26.12.2021)

SAATAVISSA: <https://github.com/tlikonen/latex-opas>

LISENSSI: *Creative Commons Nimeä-JaaSamoin 4.0 Kansainvälinen* (CC BY-SA 4.0). Lisenssi antaa sinulle luvan kopioida ja levittää tätä teosta tai sen osia missä tahansa välineessä ja muodossa. Sisältöä saa muokata, ja sen pohjalta saa luoda uusia teoksia mihin tahansa tarkoitukseen, myös kaupallisesti. Ehdot ovat seuraavat:

- Sinun on mainittava tekijä(t) asianmukaisesti, tarjottava linkki lisenssin koko tekstiin (ks. alla) sekä mainittava, mikäli olet tehnyt muutoksia.
- Jos muokkaat teosta tai luot sen pohjalta uuden teoksen, sinun on jaettava muutoksiasi samalla lisenssillä kuin alkuperäistä teosta.
- Et saa asettaa sellaisia oikeudellisia ehtoja tai teknisiä estoja, jotka estävät muita tekemästä asioita, jotka tämä lisenssi sallii.

Lisenssin koko teksti:

<https://creativecommons.org/licenses/by-sa/4.0/legalcode.fi>

Sisälllys

Esipuhe	9
1 Valmistautuminen	11
1.1 Käsitteet ja nimet	11
1.1.1 Tex ja Latex	12
1.1.2 LuaLatex ja XeLatex	12
1.1.3 Latex yläkäsitteenä	13
1.2 Asentaminen tietokoneelle	14
1.3 Apuohjelmia	15
1.3.1 Tekstieditori	15
1.3.2 Pdf-katselin	15
1.3.3 Latexmk	16
1.3.4 Texdoc	18
1.4 Lähdetiedostot	18
2 Merkintäkieli ja perustekniikka	21
2.1 Merkistö	21
2.1.1 Varatut erikoismerkit	21
2.1.2 Sanaväli	22
2.1.3 Rivinvaihto lähdedokumentissa	23
2.1.4 Kappaleen vaihtuminen	24
2.1.5 Kommentit ja muistiinpanot	24
2.1.6 Aaltosulkeet	25
2.1.7 Sitova välilyönti	25
2.1.8 Ohuke	25
2.1.9 Lainausmerkit ja heittomerkki	26

2.1.10	Yhdysmerkki, ajatusviiva ja miinusmerkki	28
2.1.11	Kolme pistettä eli ellipsi	29
2.1.12	Ylä- ja alaindeksi	29
2.1.13	Tavutusvihje	30
2.1.14	Tarkkeet ja erikoismerkit	31
2.2	Komennot	34
2.2.1	Omat komennot ja abstrahointi	35
2.2.2	Komentojen määrittely	36
2.2.3	Kestävät ja hauraat komennot	37
2.2.4	Muita vinkkejä	38
2.3	Ympäristöt	40
2.4	Mitat	42
2.4.1	Mittayksiköt	42
2.4.2	Mittakomennot ja typografinen viivasto	44
2.4.3	Venyvät mitat ja välit	46
2.5	Laskurit	48
2.5.1	Hierarkkiset laskurit	50
2.5.2	Kokonaislaskurit	51
2.6	Laatikot	52
2.6.1	Pienet laatikot	53
2.6.2	Suuret laatikot	54
2.6.3	Laatikoiden siirtely	56
3	Asetukset	59
3.1	Dokumenttiluokat	59
3.1.1	Perusdokumenttiluokat	60
3.1.2	Perusdokumenttiluokkien asetukset	61
3.1.3	Muita dokumenttiluokkia	64
3.2	Sivu	65
3.2.1	Sivun koko ja marginaalit	65
3.2.2	Sivun mittoja	68
3.2.3	Leikkuuvarat	69
3.2.4	Ylä- ja alatunnisteet	70
3.3	Pdf-tiedosto	74
3.4	Fontit	77
3.4.1	Fonttien määrittäminen	78
3.4.2	Fontin koko ja rivikorkeus	81
3.4.3	Kirjainperheen ja -leikkauksen valitseminen	84

3.4.4	Fonttikoon määrittely suhteellisesti	86
3.4.5	Fonttikoon määrittely absoluuttisesti	87
3.4.6	Fonttien oletusasetuksia	90
3.4.7	Typografiset ligatuurit	91
3.4.8	Numeroiden muoto	93
3.4.9	Välilystykset: harvennus ja tiivistys	94
3.4.10	Keinotekoinen venytys, lihavointi ja kallistus	95
3.4.11	Keinotekoinen pienversaali	96
3.4.12	Matematiikkatilan fontti	96
3.5	Kieli	98
3.5.1	Polyglossia	99
3.5.2	Babel	100
3.6	Tavutus	102
3.6.1	Yleiset tavutussäännöt	102
3.6.2	Yksittäisten sanojen tavutus	103
3.6.3	Tavutuksen sallivia ja sitovia merkkejä	104
3.6.4	Tavutus sanan reunasta	106
3.6.5	Muita tavutusasetuksia ja -vinkkejä	107
3.6.6	Suomen kielen tavutus	108
3.6.7	Tavutus pois päältä	110
4	Rakenne ja sisältö	113
4.1	Tekstikappaleet	113
4.1.1	Tasaaminen ja palstan muoto	114
4.1.2	Pystysuuntaiset välit	115
4.1.3	Ensimmäisen rivin sisennys	117
4.1.4	Riippuva sisennys	119
4.1.5	Vasen ja oikea sisennys sekä lohkolainaukset	121
4.1.6	Rivinvaihtokomennot	123
4.1.7	Lesket ja orvot	124
4.1.8	Marginaalihuomautukset	125
4.1.9	Anfangit eli suurikokoiset alkukirjaimet	127
4.2	Tekstin korostaminen	128
4.2.1	Kursiivi, kallistus ja lihavointi	129
4.2.2	Pienversaali eli kapiteeli	130
4.2.3	Alleiviivaus ja yliviivaus	132
4.2.4	Harvennus ja tiivistys	134
4.2.5	Tietokonekoodi ja tasalevyinen kirjainperhe	135
4.2.6	Verkko-osoitteet	136
4.2.7	Värit	138
4.3	Sivunvaihdot ja sivujen tasaaminen	141

4.4	Jäsennys	143
4.4.1	Perustiedot, nimiösivu ja tiivistelmä	143
4.4.2	Otsikointi	144
4.4.3	Otsikoiden numerointi	148
4.4.4	Otsikoiden ulkoasu	149
4.4.5	Esittely, pääluvut, liitteet ja luettelot	152
4.4.6	Tyypillinen tietokirjan rakenne	152
4.5	Sisällysluettelo	153
4.6	Luetelmat	157
4.6.1	Perusympäristöt	158
4.6.2	Omat luettelmat (list)	161
4.7	Taulukot	167
4.7.1	Perustoiminnot	168
4.7.2	Asetuksia	170
4.7.3	Poikkeuksellisia sarakkeita ja rivejä	171
4.7.4	Kiinteälevyiset taulukot	173
4.7.5	Muita paketteja	176
4.8	Sarkaimet	177
4.8.1	Sarkainympäristö	177
4.8.2	Laatikkototeutus	179
4.9	Leijuvat osat	180
4.9.1	Leijuvat taulukot ja kuvat	181
4.9.2	Muut leijuvat osat	183
4.9.3	Leijuvien osien luettelot	184
4.9.4	Sijoittelu sivulle	186
4.9.5	Ulkoasu	189
4.10	Ristiviittaukset	193
4.11	Alaviitteet	195
4.12	Palstat	199
4.12.1	Kahden palstan tila	199
4.12.2	Palstaympäristö	200
4.13	Lähdeluettelo ja lähdeviitteet	203
4.13.1	Peruskäyttöön (natbib)	203
4.13.2	Vaativaan käyttöön (biblatex)	207
	Teostietokanta (207) – Käyttöönotto (210) – Lähdeviittaukset (213) – Lähdetiedostojen kääntäminen (213) – Lähdeluettelon mittoja (214) – Muita asetuksia (215).	

4.14	Asiahakemistot	219
4.14.1	Asiahakemistojen määrittely	220
4.14.2	Asiasanojen lisääminen	222
4.14.3	Hakemiston latominen ja asetukset	227
4.15	Kuvat	229
4.15.1	Kuvatiedostot	229
4.15.2	Vektorigrafiikka	231
4.16	Matematiikka	235
4.16.1	Matematiikkatilan käyttö	235
4.16.2	Matematiikkatilan kielioppia	237
4.16.3	Erikoismerkkejä	241
5	Erikoisdokumentit	245
5.1	Esitysgrafiikka	245
5.1.1	Diaesityksen rakentaminen	245
5.1.2	Ulkoasuteemat	248
5.1.3	Muita asetuksia	250
5.2	Kirjeet	252
	Kirjallisuutta	255
	Asiahakemistot	257

Esipuhe

Tex ja Latex kehitettiin alun perin parantamaan tietokoneavusteista tekstin latomista. Ohjelmat syntyivät 1970-luvun lopulla ja 1980-luvun alkupuolella eli ajalla, jolloin tietokoneilla ei vielä kovin hyvin osattu tuottaa laadukasta typografiaa ja painotuotteita. Texin alkuperäinen luoja on Donald Knuth, ja Latexin kehitti Leslie Lamport.

Texin ja Latexin myötä tietokonetytografia parani: tekstieditorin ja merkintäkielen avulla kirjoittaja pystyi varsin helposti tuottamaan hyvää jälkeä niin kuin typografian ja ladonnan ammattilaiset aikoinaan. Varsinkin akateemisten tekstien tuottamisessa Latex sai vankan jalansijan. Kirjoittaja keskittyy lähinnä sisältöön, ja tietokoneohjelmat hoitavat taittamisen eli ulkoasun suunnittelun (Latex) ja lopullisen dokumentin latomisen (Tex). Typografinen ammattitaito on ohjelmoitu sisään näihin tietokoneohjelmiin.

Sitten tulivat tekstinkäsittelyohjelmat, taitto-ohjelmat ja uusi käsite WYSIWYG eli *what you see is what you get*. Suoraan tietokoneen ruudulta eli graafisesta tietokoneohjelmasta näki lopullisen painotuotteen sellaisenaan. Mikä sen helpompaa? Jokainen kirjoittaja oli nyt samassa hetkessä myös latoja, joka heittelee kirjakkeita tietokoneen ruudulle peräkkäin: sanoiksi, riveiksi, palstoiksi ja lopulta valmiiksi dokumentiksi. Tekstien parissa työskentely helpottui, mutta typografiaa tämä kehitys ei parantanut, koska yhä useammat kirjoittajat saivat vastuulleen myös ulkoasun suunnittelun ja lopullisten dokumenttien valmistuksen. Vuosisatojen aikana kehittynyt typografinen osaaminen ei enää sisällynyt tietokoneohjelmaan.

Tässä tilanteessa olemme edelleenkin: lähes jokainen kirjoittaja vastaa itse niin sisällöstä kuin typografiastakin eli lopullisesta ulkoasusta. Jokainen voi olla tekstinsä julkaisija. Visuaalisten tekstinkäsittely- ja taitto-ohjelmien yleistyminen ei kuitenkaan lopettanut Latexin tarinaa. Latexin kehitys ei jäänyt 1980-luvulle, vaan sen ympärillä tapahtuu jatkuvasti edelleenkin. Esimerkiksi nykyaikainen fonttitekniikka ja tietokoneiden laajat merkistöt ovat myös Latexin käyttäjän arkipäivää.

Jos siis typografia, laadukkaat julkaisut ja niihin liittyvä tekniikka kiinnostavat, on Latexilla varmasti paljon annettavaa nykyajallekin. Toivottavasti tästä oppaasta on apua tutustumismatkassa – ja sen jälkeenkin. Tervetuloa mukaan!

Luku 1

Valmistautuminen

Tämän pääluvun tarkoituksena on johdatella uudet ihmiset Latexin pariin. Tämän luettuasi sinun pitäisi olla valmis aloittamaan Latexin käyttö ja opiskelemaan sen tekniikkaa eteenpäin. Jos taas olet jo tottunut kääntämään Latex-lähdetiedostoja, voit aivan hyvin hypätä tämän luvun yli tai ehkä vain silmäillä tekstiä lisävinkkien toivossa.

1.1 Käsitteet ja nimet

Latex ja sen ympärille rakentuneet ohjelmistot ovat aika monimutkainen kokonaisuus, johon kuuluu eri-ikäisiä ja abstraktiotasoltaan erilaisia osia. Mukaan kuuluu tietenkin konkreettisia tietokoneohjelmia, jotka tekevät konkreettisia, ennalta määriteltyjä asioita. Mukaan kuuluu kuitenkin myös ihmisten luomia abstrakteja käsitteitä, jotka ovat vaikeammin määriteltävissä.

Internetissä näkyy silloin tällöin termi- ja käsittekeskusteluja, jossa ihmetellään, mihin mikäkin Latexiin liittyvä palikka kuuluu käsitteellisesti. Millainen suhde joillakin uudemmissa osilla on vanhempiin? Oikeiden termien ja käsitteiden osaamisesta on hyötyä ainakin silloin, kun pyytää verkossa apua suurelta yleisöltä. Viestintähän aina vaatii, että puhutaan suunnilleen samaa kieltä, joten seuraavaksi selvennetään hieman Latexiin liittyviä peruskäsitteitä.

1.1.1 Tex ja Latex

Tex on tekstin ladontaan erikoistunut ohjelmointikieli. Kielen avulla voidaan antaa ladontaohjeita ja ilmaista muuta siihen liittyvää logiikkaa. Ohjelmointikielellä kirjoitettujen ohjeiden perusteella tietokoneohjelma osaa latoa tekstidokumentin ihmisten luettavaksi. Niinpä Tex on myös tietokoneohjelma (`tex`), joka osaa lukea Tex-ohjelmointikieltä sisältävän tekstitiedoston ja tuottaa sen perusteella valmiin julkaistavan dokumentin. Yleensä Tex-ohjelmointikieltä ei kuitenkaan käytetä suoraan tekstidokumenttien toteuttamiseen. Ohjelmointikieltä kirjoittavat tavallisimmin vain ne, jotka haluavat kehittää itse ladontajärjestelmää paremmaksi muita ihmisiä varten.

Latex toimii korkeammalla abstraktiotasolla kuin Tex. Se on laaja koelma toimintoja, jotka piilottavat monimutkaiset tekniset yksityiskohdat ja tarjoavat ihmisille varsin helppokäyttöisen merkintäkielen, jolla omat tekstidokumentit voi toteuttaa. Latex-merkintäkielen kirjoittaminen ei ole ohjelmointia, vaan se on oman dokumentin sisällön, rakenteen ja ulkoasun kuvailua tietynlaisten merkintätapojen avulla.

Latex-merkintäkielellä kuvatut dokumentit välitetään tietokoneohjelmalle jatkokäsitteltäväksi. Niinpä Latex on myös tietokoneohjelma (`latex`, `pdflatex`), jolla merkintäkielinen lähdetiedosto käännetään julkaistavaksi `dvi`- tai `pdf`-dokumentiksi.¹

Latex-järjestelmästä käytetään tällä hetkellä versiota Latex 2 ϵ , joka julkaistiin jo vuonna 1994 ja johon ilmestyy pieniä parannuksia vuosittain. Perusosat ovat kuitenkin varsin muuttumattomia. Varsinainen kiinnostava kehitys tapahtuukin esimerkiksi Latexin kääntäjissä (Lua-latex ja Xelatex) sekä eri tekijöiden dokumenttiluokissa (luku 3.1) ja laajennuspaketeissa. Viimeksi mainitut laajentavat perus-Latexia tuomalla niihin lisää helppokäyttöisiä toimintoja.

1.1.2 LuaLatex ja XeLatex

Nykyaikana Latex-dokumentteja ei juuri käännetä alkuperäisillä kääntäjillä (`latex`, `pdflatex`, `tex`) vaan kehittyneemmillä kääntäjäohjelmilla.²

¹ `DVI` = device independent file format; `PDF` = portable document format.

² Englannin kielellä Latexin kääntäjiä on tapana kutsua yleisnimellä *engine* 'kone, moottori'.

Niistä tärkeimmät ovat LuaLatex ja XeLatex.³ Ne muun muassa osaavat lukea Unicode-merkistöllä kirjoitettuja lähdedokumentteja ja käyttää nykyaikaisia True Type- ja Open Type -fontteja, mitä alkuperäinen LaTeX ja TeX eivät osaa.

LuaLatexilla ja XeLatexilla ei ole ohjelmien käyttäjän kannalta suurtakaan eroa – ei välttämättä mitään näkyvää eroa. Miksi sitten on olemassa kaksi, ulkoisesti lähes samanlaista kääntäjää? Tärkein syy taitaa olla se, että XeLatex tehtiin ensin. Tarkoituksena oli saada Unicode-merkistön tuki ja fonttiasiat ajan tasalle. Myöhemmin jotkut ajattelivat, että Lua-ohjelmointikieli täytyy saada mukaan. Sillä on merkitystä joillekuille laajennuspakettien tekijöille. Lua-kielen sisällyttäminen oikeastaan pakotti kirjoittamaan koko homman uusiksi. Toteutuksissa on muitakin sisäisiä eroja. XeLatex oli pitkään suositumpi ja paremmin tuettu eri laajennuspaketeissa, mutta erot ovat sittemmin tasoittuneet.

Latex-kokeilun alkutaipaleella voi vaikka arpoa kolikon avulla, kumpaa kääntäjää käyttää, sillä niiden erot eivät ihan helposti tule esiin. Jotkin harvat laajennuspaketit eivät toimi LuaLatexissa, kun taas toiset eivät toimi XeLatexissa. Joskus saattaa jopa tulla vastaan yksittäinen, kehityksestä ehkä jälkeen jäänyt paketti, joka ei toimi kummassakaan vaan ainoastaan perinteisillä kääntäjillä (latex, pdflatex).

Joissakin tilanteissa eri kääntäjien tekemissä pdf-dokumenteissa näkyy pieniä eroja. Ainakin yksittäisissä fonttien asetuksissa ja joidenkin Unicode-merkkien käsittelyssä on eroja. Tämänkin oppaan tekstissä mainitaan pari pientä ominaisuutta, jotka toimivat vain toisella kääntäjällä: toiset LuaLatexilla, toiset XeLatexilla.

1.1.3 Latex yläkäsitteenä

Jotta kaikki olisi mahdollisimman sekavaa, ”Latex” toimii myös yleisnimityksenä tälle kaikelle. Se esiintyy ilmauksissa kuten ”Toteutin dokumentin LaTeXilla” tai ”Tämä artikkeli on tehty LaTeXilla”. Ilmaukset sitten tarkoittavat suunnilleen seuraavanlaista: Henkilöllä on asennettuna tietokoneelle Latex-jakelukokonaisuus (kuten TeX Live). Hän on kirjoittanut tekstieditorilla (kuten GNU Emacsilla) tekstitiedoston, jossa

³ Tietokoneohjelmat luaLatex ja luatex sekä xeLatex ja xetex.

on dokumentin sisältö ja Latex-merkintäkielisiä komentoja mutta ehkä myös joitakin Tex-komentoja. Sitten hän on kääntänyt eli ladottanut tekstitiedostonsa pdf-tiedostoksi Latex-ladontaohjelman jollakin toteutuksella kuten Lualatexilla tai Xelatexilla.

Meille taitaa riittää vain Latexista puhuminen, mutta siitäkin on mainittava vielä yksi asia. Latexin harrastajat tykkäävät käyttää dokumenttiansa leipätekstissä ladontajärjestelmän logoja kuten \TeX ja \LaTeX . Usein teksteissä näkyy myös logojen pohjalta mukailtuja kirjoitusasuja TeX ja LaTeX.

Kielenhuoltajien suositusten mukaan logojen eikä erikoisten kirjotusAsujen paikka ei ole asiatyylisen tekstilajien leipätekstissä. Nimet ovat osa kielen järjestelmää ja käyttäytyvät normaalissa tekstissä sen mukaisesti. Niinpä tässä oppaassa käytetään erisnimiä kielenhuollon normien mukaisesti, esimerkiksi Tex ja Latex. Koodi ja komennot ovat siinä muodossa kuin ne tietokoneelle annetaan, esimerkiksi lualatex. Tasalevyinen, kirjoituskonetyylinen fontti on merkkinä siitä, että kyse on tietokonekoodista.

1.2 Asentaminen tietokoneelle

Latex pitää tietysti asentaa tietokoneelle, jotta sitä voisi käyttää. Miten edellisessä luvussa kuvattu sekava kokonaisuus saadaan ehjänä omalle tietokoneelle? Onneksi muut ovat jo ratkaisseet sen ongelman.

Tavallisin tapa Latexin käyttöönottoon on jonkin Latexin jakelupaketin asentaminen. Jakelupaketti sisältää Latexin perusosien lisäksi paljon laajennuspaketteja ja niiden ohjekirjoja. Kaikkea ei koskaan tarvitse, mutta kun yllättävä tarve tulee tai lukee vinkkejä verkkokeskusteluista, on mukavaa huomata, että paketti olikin itsellä jo valmiina. Siksi kokonaisen jakelupaketin asentaminen on helpoin tapa.

GNU/Linuxissa ja muissa Unix-tyyppisissä käyttöjärjestelmissä käytetään yleensä Tex Live -nimistä jakelua. Se on todennäköisesti saatavilla käyttöjärjestelmäjakelun pakettivarastoista. Esimerkiksi Debianiin⁴

⁴ <https://www.debian.org/>

ja sen kaltaisiin järjestelmiin on asennuspaketti ”texlive-full”, joka asentaa kaiken helposti ja kerralla.

Windows-käyttöjärjestelmälle on saatavilla Tex Liven lisäksi Miktex ja Protext. Mac os -käyttöjärjestelmän kanssa käytettäneen yleensä Mactex-nimistä jakelua.

1.3 Apuohjelmia

1.3.1 Tekstieditori

Omien dokumenttien ja Latex-merkintäkielen kirjoittamiseen kannattaa käyttää kunnollista tekstieditoriä, koska se on tärkein työkalu ja sen kanssa ollaan eniten tekemisissä. Pyri löytämään sellainen editori, joka osaa värjätä tekstiä Latexin tai Texin tekstipiirteiden mukaisesti. Väreillä ei sinänsä ole merkitystä, mutta editorin laadusta se yleensä kertoo paljon. Jos editori tuntee erilaisten ohjelmointi- ja merkintäkielten luonnetta ja osaa merkitä kielen avainsanoja havainnollisilla väreillä, se todennäköisesti on tehty tehokkaaseen ohjelmointiin ja muuhun vastaavaan työskentelyyn. Ihan yksinkertaisiin editoreihin ei tuollaisia ominaisuuksia yleensä tehdä.

1.3.2 Pdf-katselin

Latex-kääntäjät eli -moottorit kuten LuaLatex ja XeLatex tuottavat pdf-tiedoston, ja niiden katselemiseen tarvitaan tietenkin oma ohjelmansa. Sellaisia on saatavilla paljon erilaisia, ja melkein mikä tahansa kelpaa, mutta yksi tietty ominaisuus olisi toivottavaa olla: muuttuneen pdf-tiedoston automaattinen lataaminen.

Välillä työskentely on sitä, että tehdään Latex-dokumenttiin pieni muutos, käännetään se ja katsotaan pdf:ää. Lopputulos ei ehkä ihan miellytä. Muokataan tekstiä tai asetuksia vähän, käännetään ja katsotaan, miltä ladottu pdf nyt näyttää.

On suuri apu, jos pdf-katselimessa ei tarvitse joka kerta valikoiden kautta avata samaa tiedostoa uudelleen, vaan ohjelma itse huomaa, että jo avattu tiedosto muuttui tiedostojärjestelmässä, ja lataa sen automaattisesti uudelleen. Jotkin pdf-ohjelmat osaavat tämän. Jotkin ohjelmat

eivät ihan osaa mutta osaavat sentään yhdellä näppäinpainalluksella avata saman pdf:n uudelleen tiedostojärjestelmästä.

Hyvän tekstieditorin ja pdf-katselimen kanssa työskentely on sujuvaa. Parhaimmillaan editorissa tietty näppäinkomento tallentaa ja kääntää dokumentin, ja pian pdf-katselin lataa muuttuneen pdf:n automaattisesti näkyviin. Sekä editorin että pdf-katselimen voi pitää esillä samanaikaisesti.

1.3.3 Latexmk

Erinomaisen hyödyllinen apuohjelma on Latexmk, koska se helpottaa dokumenttien kääntämistä ja muutakin työskentelyä. Varsin usein Latex-dokumentit täytyy kääntää useita kertoja ennen kuin pdf-tiedosto on valmis. Tämä johtuu siitä, että dokumentit sisältävät usein ristiviitteitä eli viittauksia dokumentin toisiin osiin. Latex ei saa ristiviitteitä kohdalleen yhdellä kääntämisellä, vaan ensin se kirjoittaa viittauskohteet muistiin väliaikaistiedostoon ja seuraavilla kääntökerroilla käyttää väliaikaistiedostoa apunaan.

Tavallinenkin Latexin kääntäjä kyllä huomauttaa tietokoneen käyttäjää, kun uusintakäännös on tarpeen, mutta Latexmk-ohjelma käynnistää uusintakäännöksen itse, aina kun se on tarpeellista. Alla ovat esimerkkikomennot Latex-dokumentin kääntämiseen Lualatexilla ja Xelatexilla.

```
latexmk -lualatex teksti.tex  
latexmk -xelatex teksti.tex
```

Työskentelyä erityisen paljon helpottava valitsin on `-pvc`. Kun tuo valitsin on mukana komennossa, Latexmk jää tarkkailemaan annettua Latex-tiedostoa, ja kun se huomaa tiedoston muuttuneen, se kääntää tiedoston automaattisesti uudelleen. Kirjoittajan ei siis tarvitse muuta kuin tallentaa tiedosto tekstieditorista, ja tarkkailutilassa oleva Latexmk kääntää sen aina itsestään.

Muitakin hyödyllisiä toimintoja on mukana. Seuraavista esimerkeistä ensimmäinen komento poistaa kääntämisen aikana luodut väliaikais-


```
1 $pdf_mode = 4; # 4=lualatex, 5=xelatex
2 $lualatex = 'lualatex -interaction=nonstopmode -shell-escape %O %S';
3 $xelatex = 'xelatex -interaction=nonstopmode -shell-escape %O %S';
4 $clean_ext = 'smn nav xdv bbl run.xml';
5 $pdf_previewer = 'okular %S';
```

Esimerkki 1.1: Latexmk-ohjelman asetustiedosto (~/.config/latexmk/latexmkrc).

tiedostot⁵, ja jälkimmäinen komento poistaa kaikki luodut tiedostot eli väliaikaistiedostojen lisäksi myös valmiin pdf-tiedoston.

```
latexmk -c teksti.tex
latexmk -C teksti.tex
```

Edellisissä esimerkeissä käsitellään lähdetiedostoa nimeltä teksti.tex, mutta jos lähdetiedostoa ei anna komennolle lainkaan, käännetään kaikki nykyisessä hakemistossa olevat tex-päätteiset tiedostot.

Latexmk-ohjelmalle voi tehdä asetustiedoston, johon voi kirjoittaa omaan käyttöön sopivat asetukset. Asetustiedosto sijoitetaan tiedostojärjestelmässä käyttäjän kotihakemistoon. Esimerkki 1.1 näyttää, mitä se voisi ehkä sisältää.

Esimerkin ensimmäisen rivin asetus määrittää, miten pdf-tiedostot tuotetaan tai mitä kääntäjää käytetään oletuksena. Toisella ja kolmannella rivillä määritellään, millä tavoin Lualatex ja Xelatex suoritetaan. Tässä esimerkissä oletusasetuksiin on lisätty `-interaction=nonstopmode`, joka estää kaiken vuorovaikuttaisen toiminnan. Asetus on tarpeen ainakin silloin, kun kääntäjä käynnistetään toisesta ohjelmasta kuten tekstieditorista eikä vuorovaikutus kääntäjän kanssa ole mahdollista. Valitsin `-shell-escape` kytkee päälle ominaisuuden, jota tarvitaan joidenkin laajennuspakettien toimintaan.⁶

Esimerkin 1.1 neljännellä rivillä luetellaan kääntämisen aikana syntyvien väliaikaistiedostojen päätteitä. Latexmk-ohjelma tuntee yleisimmät väliaikaistiedostot (`log`, `aux`, `out` ym.), mutta tällä asetuksella mu-

⁵ Kääntäjän luomien väliaikaistiedostojen nimien päätteitä: `log`, `aux`, `out` ym.

⁶ Ainakin asiahakemistopakettit `indextools` ja `imakeidx` tarvitsevat `-shell-escape`-toiminnon (luku 4.14).

kaan voi lisätä harvinaisempia, joita se ei tunne. Viides rivi määrittää pdf-katseluohjelman, joka käynnistetään, kun käytetään valitsimia `-pv` tai `-pvc`.

1.3.4 Texdoc

Latexin kirjoittajan täytyy silloin tällöin lukea ohjekirjoja. Vaikka Latexin perusosat joskus oppisikin ulkoa, ei voi koskaan muistaa kaikkien hyödyllisten laajennuspakettien kaikkia ominaisuuksia.

Tex Live -jakelun (luku 1.2) mukana tulee mainio komentotulkissa toimiva komento `texdoc`, jolla voi hakea ja avata omaan järjestelmään asennettuja Latex-aiheisia ohjeita. Jos vaikka haluaa tutustua esimerkiksi 1.2 (s. 19) mainittavaan `fontspec`-pakettiin syvällisemmin, tarvitsee vain komentaa `texdoc fontspec`, ja paketin pdf-muotoinen ohjekirja avautuu.

1.4 Lähdetiedostot

Seuraavassa ovat ohjeet alkuun pääsemiseksi. Luodaan yksinkertainen Latex-lähdetiedosto, jota voi käyttää harjoitteluun ja pohjana omille töille. Tallenna esimerkin 1.2 sisältö tekstieditorin avulla tiedostoon vaikkapa nimellä `teksti.tex`. Käännä eli lada se pdf-tiedostoksi käyttämällä jotakin seuraavista komennoista (valitse yksi):

```
lualatex teksti.tex
xelatex teksti.tex
latexmk -lualatex teksti.tex
latexmk -xelatex teksti.tex
```

Tuloksena pitäisi olla tiedosto `teksti.pdf`, jota voi ihailia jollakin pdf-tiedostojen katseluun tarkoitettulla ohjelmalla.

Esimerkin 1.2 ensimmäisellä rivillä määritellään dokumenttiluokka `article`, joka on tietynlainen sivupohja tai asetusten kokoelma, jonka perustalle aletaan rakentaa omaa dokumenttia. Luokka `article` on tyypillinen valinta lyhyehköille teksteille. Lisätietoa dokumenttiluokista on luvussa 3.1.

Riveillä 2–4 käytetään komentoa `\usepackage`, jonka avulla otetaan käyttöön sivun asetuksista huolehtiva `geometry`-paketti, fonttiasetuksia

```

1 \documentclass{article}
2 \usepackage[a4paper, top=20mm, bottom=30mm, left=20mm, right=20mm]{geometry}
3 \usepackage{fontspec}
4 \usepackage{polyglossia}
5
6 \setdefaultlanguage{finnish}
7 \setmainfont{Latin Modern Roman}[Scale=1.3]
8 \linespread{1.4}
9
10 \begin{document}
11
12 Minun Latex-dokumenttini!
13
14 \end{document}

```

Esimerkki 1.2: Latex-lähdedokumentin runko ja perusasetukset.

hoitava `fontspec`-paketti ja kieliasetuksista vastaava `polyglossia`-paketti. Näitä kolmea tarvitaan melkein joka kerta dokumenteissa, ja niihin palataan tarkemmin luvuissa 3.2, 3.4 ja 3.5.

Seuraavilla riveillä asetetaan kieleksi suomi (`finnish`) ja määritetään oletuksena käytettävä fontti tai oikeastaan kokonainen kirjainperhe. `Latin Modern Roman` -kirjainperheen tilalle voi toki asettaa jonkin muunkin. Fontin oletuskoko on 10 typografista pistettä, mutta tässä esimerkissä se venytetään 1,3-kertaiseksi eli 13 pisteeseen. Riviväliin liittyvä kerroin asetetaan rivillä 8.

Dokumentin alkuosaa riville 9 saakka kutsutaan esittelyosaksi (`preamble`). Tässä osassa ladataan tarvittavat paketit ja määritetään dokumentin asetuksia ja taustatietoja. Riviltä 10 alkaa varsinainen tekstiosa eli dokumentin sivuille ladettava sisältö. Se osa kirjoitetaan `document`-ympäristön sisään eli riveillä 10 ja 14 olevien ympäristön aloitus- ja lopetuskomentojen väliin (`\begin`, `\end`).

Tällaisen merkintäkielen avulla dokumentit kirjoitetaan. Osa merkintäkielen komennoista tulee Latexin perusosasta ja osa tulee erikseen ladattavista paketeista (`geometry`, `fontspec`, `polyglossia` ym.). Komentoja voi luoda itsekin.

Myöhempää käyttöä varten voisi olla hyödyllistä tallentaa tämänkaltaisen pohjadokumentti. Välttyy samojen perusjuttujen kirjoittamiselta, kun voi aloittaa työt valmiista dokumenttipohjasta.

Lähdetiedoston nimissä kannattaa pitää melko suppeassa merkivalikoimassa, ja varsinkin välilyöntejä kannattaa välttää. Nimittäin kääntämisen aikana Latex ja sen paketit luovat väliaikaistiedostoja, joilla sama nimen osa kuin lähdetiedostossa, ja näitä tiedostoja saattavat käsitellä monet erilaiset taustalla vaikuttavat työkaluohjelmat. Tiedoston nimissä olevat välilyönnit ja ehkä muutkin erikoisemmat merkit aiheuttavat ongelmia.

Pitkä lähdedokumentti voi olla mielekästä jakaa useammaksi tiedostoksi. Yhteen lähdetiedostoon voi sisällyttää toisen tiedoston käyttämällä `\input`-komentoa. Komennon argumentiksi annetaan ladattavan lähdetiedoston nimi:

```
\input{toinen.tex}
```

Hieman vastaava komento on `\include`, joka myös lisää automaattisen sivunvaihdon (`\clearpage`, luku 4.3) komennon kohdalle.

Nyt lienee sopiva aika alkaa opiskella itse Latexia eli merkintäkieltä ja kokeilla sen ominaisuuksia itse. Tätä opasta ei tarvitse lukea järjestyksessä luku luvulta eteenpäin, vaan eri aiheita voi vapaasti opiskella mielenkiinnon ja tarpeiden mukaan. Onnea matkaan!

Luku 2

Merkintäkieli ja perustekniikka

Latex on merkintäkieli, mikä tarkoittaa, että se sisältää omat tapansa dokumentin rakenteen ja sisällön kuvaamiseen. Kaikkea ei kirjoiteta lähdedokumenttiin sellaisenaan, vaan täytyy käyttää tiettyjä kielen sääntöjen mukaisia merkintätapoja tai komentoja.

Tässä luvussa käsitellään merkintäkielen perusasioita, joita on tarpeen ymmärtää ennen kuin voi tehokkaasti toimia Latexin parissa. Kaikkea ei tarvitse opetella ulkoa, mutta tähän lukuun on hyvä palata välillä kertaamaan perustekniikkaa.

2.1 Merkistö

Latex-lähdetiedostoon voi kirjoittaa tekstiä Unicode-merkistöllä ja sen UTF-8-koodauksella, jos kääntäjänä on Unicoden osaava ohjelma kuten LuaLatex tai XeLatex. Pääasiassa siis merkit kirjoitetaan sellaisenaan lähdetiedostoon, mutta on kuitenkin monenlaisia poikkeuksia, ja niitä käsitellään tässä alaluvussa.

2.1.1 Varatut erikoismerkit

Muutamalla merkillä on perus Latexissa erikoismerkitys, eikä niitä voi käyttää normaalilla tavalla. Merkit ovat seuraavat:

Merkki	Kirjoittaminen
%	\%
\$	\\$ \textdollar
^	\^{} \textasciicircum
_	_ \textunderscore
#	\#
&	\&
{	\{ \textbraceleft
}	\} \textbraceright
~	\~{} \textasciitilde
\	\textbackslash

Taulukko 2.1: Varattujen erikoismerkkien kirjoittaminen.

% \$ ^ _ # & { } ~ \

Useimmat näistä merkeistä voi suojata erikoismerkitykseltään kirjoittamalla niiden eteen kenoviivan (\). Tildeä (~), sirkumfleksiä (^) eikä kenoviivaa itseään ei voi suojata pelkän kenoviivan avulla, koska kenoviivan kanssa ne muodostavat eräitä muita komentoja. Taulukossa 2.1 on koottuna, kuinka edellä mainitut erikoismerkit suojataan eli saadaan ladottua dokumenttiin sellaisenaan.

Jotkin paketit määrittelevät muitakin erikoismerkkejä. Esimerkiksi kieliasetuksiin (luku 3.5) liittyvät `polyglossia-` ja `babel-` paketit voivat määritellä pari lainausmerkillä (") alkavaa, tavutuksen hallintaan liittyvää komentoa tai erikoismerkkiä.

2.1.2 Sanaväli

Välilyönti, sarkainmerkki ja yksi rivinvaihto ovat kaikki tavallisia sanavälejä Latex-dokumentissa, ja näillä kolmella on sama merkitys. Esimerkiksi rivin lopussa oleva rivinvaihto tarkoittaa samaa kuin sanojen välissä oleva välilyönti. Välilyöntejä ja sarkainmerkkejä voi kirjoittaa useita peräkkäin, mutta ne ovat sama asia kuin yksi väli.

Nämä kaikki ovat vain sanoja peräkkäin ja kuuluvat samaan kappaleeseen.

⇒ Nämä kaikki ovat vain sanoja peräkkäin ja kuuluvat samaan kappaleeseen.

Sanavälien leveys ei ole vakio. Silloin kun tekstipalsta tasataan molemmista reunoista – kuten tämänkin oppaan leipätekstissä –, rivillä olevia sanavälejä venytetään sopivasti, jotta tekstipalstan molemmat reunat saadaan tasaisiksi.

Sanavälit eivät kuitenkaan veny loputtomasti, ainakaan oletusasetuksilla, koska kovin suuret sanavälit olisivat tekstikappaleessa rumia. Jos sanaväleille haluaa antaa ”häätätilanteissa” lisää venymisvaraa, täytyy käyttää mitta `\emergencystretch`, jota käsitellään tarkemmin tekstikappaleiden yhteydessä luvussa 4.1. Mittoihin liittyvää tekniikkaa käsitellään puolestaan luvussa 2.4.

Matalatasoinen sanavälejä ja niiden venymistä säätelevä mitta on `\spaceskip`, johon voi asettaa haluamansa leveyden ja mahdolliset venymisen rajat. Tätä mitta ei ole suositeltavaa käyttää tavallisen tekstin kanssa, mutta se sopii tilanteisiin, joihin tarvitaan hyvin poikkeukselliset sanavälit.

```
\setlength{\spaceskip}{0.8em plus 0.3em minus 0.2em}
```

Myös kirjainperheelle voi asettaa oman sanavälikertoimensa fontin asetusten `WordSpace`-valitsimella. Tätä asetusta käsitellään fonttien yhteydessä luvussa 3.4.9.

2.1.3 Rivinvaihto lähdedokumentissa

Lähdedokumentissa olevat rivinvaihdot tulkitaan vain sanaväleiksi eivätkä ne vaihda riviä lopullisessa dokumentissa. Jos ladottuun dokumenttiin tarvitaan rivinvaihto, kirjoitetaan lähdedokumenttiin kaksi kenoviivaa (`\`). Tätä komentoa käsitellään tarkemmin tekstikappaleiden yhteydessä luvussa 4.1.6.

On kuitenkin mahdollista saada myös lähdedokumentin rivinvaihdot toteutumaan automaattisesti ladotussa tekstissä. Sellainen tila kytetään päälle komennolla `\obeycr`; normaaliin tilaan palataan taas komennolla `\restorecr`. Näistä komennoista voi olla hyötyä väliaikaisesti ja erityistilanteissa, mutta pysyväksi koko dokumentin tilaksi `\obeycr` ei yleensä sovi.

2.1.4 Kappaleen vaihtuminen

Tyhjä rivi lähdetiedostossa tarkoittaa kappaleen vaihtumista. Rivi on tyhjä silloin, kun se ei sisällä mitään muuta kuin rivinvaihdon tai kun se sisältää vain välilyöntejä tai sarkainmerkkejä ja lopuksi rivinvaihdon. Tyhjiä rivejä voi olla useita peräkkäin, mutta ne tarkoittavat samaa kuin yksi tyhjä rivi. Uuden tekstikappaleen voi aloittaa myös komennolla `\par`.

```
Nämä rivit kuuluvat  
samaan kappaleeseen.
```

```
Tässä on toinen tekstikappale.  
Nyt ei oteta kantaa siihen, miten  
rivit ja kappaleet muotoillaan.
```

Ladotuissa teksteissä uuden tekstikappaleen alkaminen ilmaistaan usein sisennetyllä rivillä, mutta sisennyksiä eikä muitakaan muotoiluja ei tehdä tekstieditorissa välien avulla. Kappaleiden muotoiluun on omat keinonsa, ja niistä käsitellään luvussa 4.1.

2.1.5 Kommentit ja muistiinpanot

Latex-dokumentissa prosentti merkki (%) on kommenttimerkki, jonka jälkeisen rivinosan kääntäjä jättää huomioimatta. Merkki on tarkoitettu kirjoittajan omien kommenttien ja muistiinpanojen kirjoittamiseen.

```
% Nyt ei tosin ole  
% mitään kommentoitavaa.
```

Kommenttimerkki vaikuttaa kääntäjään myös siten, että se syö kaikki välilyönnit ja sarkainmerkit, jotka tulevat kyseisen kommentin jälkeen. Tämän vuoksi kommenttimerkin avulla voi yhdistää eri riveillä olevan tekstin. Seuraava esimerkki tuottaa ladottuna ehjän sanan *Latex*:

```
La% Nämä rivit  
t% yhdistyvät.  
ex
```

⇒ Latex

2.1.6 Aaltosulkeet

Aaltosulkeet `{ }` muodostavat eräänlaisen näkymättömän ympäristön, jonka sisällä voi olla väliaikaisesti voimassa erilaiset asetukset kuin ulkopuolella. Aaltosulkeiden sisällä suoritettavat komennot, uusien komentojen määrittelyt (luku 2.2) tai asetetut mittojen arvot (luku 2.4) ovat voimassa vain kyseisen ympäristön sisäpuolella. Seuraavassa esimerkiksi aaltosulkeilla rajataan kursivointikomennon `\itshape` vaikutus- aluetta.

```
tavallinen {\itshape kursiiivi} tavallinen
```

⇒ tavallinen *kursiivi* tavallinen

2.1.7 Sitova välilyönti

Sitova välilyönti on samanlainen tyhjä merkki kuin tavallinenkin välilyönti, mutta rivinvaihtoa ei sallita sen kohdalta. Sitovalla välilyönnillä kannattaa estää esimerkiksi pienistä osista koostuvan ilmauksen hajoaminen eri riveille (esimerkki: *osa 5*). Latexissa sitova välilyönti saadaan joko tildemerkillä (`~`) tai nimenomaan siihen tarkoitettulla merkillä, jonka Unicode-tunnus on `U+00A0 NO-BREAK SPACE`.

Nämä kaksi eri merkkiä, tilde ja `U+00A0`, toimivat hieman eri tavoin. Molemmat estävät rivinvaihdon, mutta tildemerkki sallii välin venymisen samalla tavalla kuin tavallinenkin sanaväli sallii (luku 2.1.2). Sen sijaan merkki `U+00A0` on vakiolevyinen eikä siis veny muiden sanavälien tavoin. Merkkiä `U+00A0` täytyy käyttää ainakin vuorosanaviivan (`-`) ja sitä seuraavan sanan välissä, koska se väli ei saa venyä.

2.1.8 Ohuke

Ohuke on tavallista sanaväliä kapeampi väli, ja se tehdään komennoilla `\,` (kenoviiva ja pilkku). Ohukkeen leveys Latexissa on $\frac{1}{4}$ typografinen neliön leveydestä eli em-mitasta (luku 2.4). Ohuke on tasalevyinen ja sitova, eli se ei veny muiden sanavälien tavoin, ja se estää rivinvaihdon. Siksi ohuke sopii esimerkiksi pitkien lukujen ja puhelinnumeroiden ryhmittelyyn paremmin kuin sanaväli.

```
12\,750\,000
```

```
J.\,R.\,R. Tolkien
```

	Luku	Päiväys	Nimi
Sanaväli	12 750 000	9-5-2020	J. R. R. Tolkien
Ohuke	12 750 000	9. 5. 2020	J. R. R. Tolkien
Yhteen	12750000	9.5.2020	J.R.R. Tolkien

Taulukko 2.2: Sanavälin, ohukkeen ja yhteen kirjoittamisen vertailu. Suomen kielen vastaiset kirjoitusasut on viivattu yli.

Myös henkilön etunimen alkukirjainten välissä voi käyttää ohuketta, jos tavallinen sanaväli vie kirjaimet turhan kauas toisistaan. Sukunimi erotetaan kuitenkin aina sanavälillä. Joskus myös päiväyksissä käytetään ohuketta järjestysluvun pisteiden jälkeen. Taulukossa 2.2 vertaillaan sanaväliä, ohuketta ja yhteen kirjoittamista.

2.1.9 Lainausmerkit ja heittomerkki

Suomalaisessa näppäinasettelussa SHIFT eli vaihtonäppäin ja 2 tuottaa yleislainausmerkin eli niin sanotun ASCII-lainausmerkin ("), mutta se ei taida olla minkään kielen varsinainen lainausmerkki. On siis syytä käyttää oikeita lainausmerkkejä, ja se käy Latexissa varsin helposti.

Eri kielissä lainausmerkkikäytännöt ovat erilaiset. Suomen kielessä käytetään ”tällaisia” lainausmerkkejä ja joskus »tällaisia» kulmalainausmerkkejä. Jos lainauksen sisään tarvitaan lainaus, täytyy sisempi lainaus kirjoittaa ’tällaisten’ puolilainausmerkkien avulla. Yksittäin käytettynä se on nimeltään heittomerkki. Englannin kielessä lainauksen alussa ja lopussa on erilainen merkki, ja “tässä” siitä esimerkki. Samoin on puolilainausmerkin kohdalla: ‘näin’.

Latexissa voi käyttää Unicode-merkistöä ja lähdedokumenttiin voi kirjoittaa suoraan ne lainausmerkit, jotka halutaan ladottavaksi, mutta edellä mainituille merkeille on myös omat merkintätapansa. Näppäimistöltä kirjoitettava yleisheittomerkki (') tuottaa ladottuna automaattisesti oikean kaarevan heittomerkin ('). Kun kirjoittaa kaksi heittomerkkiä peräkkäin (' '), on lopputuloksena yksi kaareva lainausmerkki ("). Kahdella suurempi kuin -merkillä (>>) saadaan kulmalainausmerkki (»).

```
'Lainaus, jonka 'sisällä' on lainaus.' \\  
>>Lainaus, jonka 'sisällä' on lainaus.>>
```

⇒ ”Lainaus, jonka ’sisällä’ on lainaus.”
»Lainaus, jonka ’sisällä’ on lainaus.»

Edellä mainitut riittävät suomen kieleen, mutta englantia ja muita kieliä varten tarvitaan myös toisinpäin oleva merkki (“), joka tehdään kahdella gravisaksentilla (` `). Vastaava puolilainausmerkki (‘) tehdään yhdellä aksentilla (`). Joissakin kielissä käytetään erilaisia kulmalainausmerkkejä lainauksen alussa ja lopussa. Vasemmalle osoittava merkki («) tehdään kahdella pienempi kuin -merkillä (<<).

Joskus todella halutaan latoa yleislainausmerkki (") tai yleisheitto-merkki ('). Ne saadaan komennolla `\textquotedbl` ja `\textquotesingle`. Yksittäinen gravisaksentti (‘) tehdään komennolla `\`{}`. Lainausmerkkien merkintätapoja ja komentoja on koottu taulukkoon 2.5 (s. 32). Toisaalta kielikohtaiset asetukset (luku 3.5) voivat tuoda mukanaan myös kielikohtaisia keinoja lainausmerkkien kirjoittamiseen.

Edellä kuvatut Latexin omat lainausmerkkien merkintätavat ('', >> ym.) eli niin sanotut Tex-ligatuurit voi kytkeä päälle ja pois päältä fontin asetuksista eli `fontspec`-pakettiin kuuluvien toimintojen avulla. Fontteissa on yleensä oletuksena päällä Tex-ligatuurit eli asetetus `Ligatures=TeX`, mutta sen saa poistettua asetuksella `Ligatures=TeXReset`. Asetusta muutetaan kirjainperheen määrittelyn yhteydessä tai väliaikaisesti komennolla `\addfontfeatures`.

```
{\addfontfeatures{Ligatures=TeXReset} `` ' ' >> ' }
```

⇒ `` " >> '

Tasalevyisessä fontissa Tex-ligatuurit eivät ole päällä oletuksena, joten yksittäisissä sanoissa tai lyhyissä ilmauksissa voi estää Tex-ligatuurit esimerkiksi komennolla `\texttt`. Samalla tietysti fonttikin vaihtuu tasalevyiseksi. Fontteja käsitellään tarkemmin luvussa 3.4.

```
\texttt{`` ' ' >> ' }
```

⇒ `` ' ' >> '

Paketti `csquotes`¹ sisältää lainausmerkkeihin liittyviä komentoja ja kielikohtaista logiikkaa. Paketissa olevan `\enquote`-komennon avulla voi

¹ <https://www.ctan.org/pkg/csquotes>

jättää paketin huoleksi, miten aloittava ja lopettava lainausmerkki tai ulommat ja sisemmät lainausmerkit kirjoitetaan missäkin kielessä. Kie-
lipaketti **polyglossia** tai **babel** täytyy olla ladattuna.

```
\usepackage{polyglossia} \setdefaultlanguage{finnish}
\usepackage[autostyle=true]{csquotes}
% ...
\enquote{Lainauksen \enquote{sisällä} lainaus.}
```

⇒ ”Lainauksen ’sisällä’ lainaus.”

2.1.10 Yhdysmerkki, ajatusviiva ja miinusmerkki

Yhdyssanan osien välissä käytettävä yhdysmerkki on Latexissa taval-
linen näppäimistöltä saatava yleisyhdysmerkki (-). Merkillä on vaiku-
tusta myös sanan tavutukseen, josta on tarkempaa tietoa luvussa 3.6.

Ajatusviivaa tarvitaan esimerkiksi äärikohtien (27–29, Oulu–Rova-
niemi), luettelien, vuorosanojen ja virkkeen irrallisen lisäysten merkit-
seminen. Suomen kielessä käytetään yleensä vain lyhyttä ajatusviivaa
(-), joka tehdään Latexissa kahdella peräkkäisellä yhdysmerkillä (--).
Pitkä ajatusviiva (—) tehdään kolmella yhdysmerkillä (---). Ajatusvii-
vat vaikuttavat sanan tavutukseen samoin kuin yhdysmerkki.

```
Oulu--Rovaniemi-yhteys
```

⇒ Oulu–Rovaniemi-yhteys

Myös Unicoden ajatusviivamerkit U+2013 EN DASH ja U+2014 EM DASH
toimivat, mutta tavutuksen kannalta ne ovat käyttäytyneet eri tavoin
Lualatex- ja Xelatex-kääntäjillä. Yhteensopivuussyistä on parasta tehdä
ajatusviivat Latexin omilla merkintätavoilla eikä Unicode-merkeillä.

Silloin kun todella täytyy latoa kaksi tai kolme peräkkäistä yhdys-
merkkiä, voi käyttää tasalevyistä fonttia (`\texttt{--}`), joka oletuksena
kytkee pois Latexin ajatusviivatoiminnon. Saman asetuksen saa kyllä
mihin tahansa fonttiin, kun poistaa fontista niin sanotut Tex-ligatuurit
asetuksella `Ligatures=TeXReset`. Väliaikaisesti asetus tehdään seuraa-
vasti:

```
{\addfontfeatures{Ligatures=TeXReset} -- ---}
```

⇒ -- ---

Miinusmerkille (–) ei Latexissa ole erityistä merkintätapaa muuten kuin matematiikkatilassa (luku 4.16). Tavallisessa tekstitulossa lyhyttä ajatusviivaa voi ja saa käyttää myös miinusmerkkinä, mutta vielä parempi olisi käyttää varsinaista Unicoden miinusmerkkiä U+2212 MINUS SIGN, koska se on fonteissa suunniteltu tyypografisesti yhteensopivaksi muiden matemaattisten merkkien kanssa.

2.1.11 Kolme pistettä eli ellipsi

Ajatuksen katkeamista ja muuta sellaista ilmaisevalle kolmelle pisteelle eli ellipsille (...) on oma merkkinsä, ja fontissa se saattaa näyttää hieman erilaiselta kuin kolme peräkkäistä pistemerkkiä. Tyypillisesti ellipsimerkissä pisteet ovat hieman harvemmassa ja erottuvat toisistaan paremmin kuin kolmena erillisenä merkinä ladotut pisteet. Ellipsi tehdään Latexissa komennoilla `\dots`, `\ldots`, `\textellipsis` tai Unicode-merkillä U+2026 HORIZONTAL ELLIPSIS.

2.1.12 Ylä- ja alaindeksi

Yläindeksit (a^2) tehdään komennolla `\textsuperscript` ja alaindeksit (a_2) komennolla `\textsubscript`.

```
a\textsuperscript{2} a\textsubscript{2}
```

Oletusasetuksilla Latex toteuttaa indeksit mekaanisesti pienentämällä fonttia ja sijoittamalla pienennetyt tekstin alas peruslinjan tuntuun tai ylös gemenalinjan yläpuolelle.² Lopputulos ei ole tyypografisesti välttämättä kovin hyvä, koska fontin pienentäminen ohentaa samalla merkkien viivoja ja ohuimmat hiusviivat voivat lähes kadota.

Open Type -fontit sisältävät usein tuen oikeille ylä- ja alaindeksille, jotka fontin suunnittelija on toteuttanut. Niitä kannattaa käyttää, koska suunnittelija tuntee oman fonttinsa ja on saanut todennäköisesti parempaa jälkeä kuin Latex mekaanisesti. Open Type -fonttien indeksit on käteväntä ottaa käyttöön `realscripts`³-paketin avulla.

Paketti `realscripts` määrittelee uudelleen Latexin ylä- ja alaindeksikomennot, niin että ne ensisijaisesti pyrkivät käyttämään Open Type -fon-

² Katso typografinen viivasto eli kuva 2.1 sivulla 45.

³ <https://www.ctan.org/pkg/realscripts>

Komento	Open Type	Mekaaninen
<code>\textsuperscript</code>	X^{ab36}	X^{ab36}
<code>\textsubscript</code>	H_2SO_4	H_2SO_4

Taulukko 2.3: Ylä- ja alaindeksien vertailua. Oikeat Open Type -fonttien indeksit saadaan `realscripts`-paketin avulla. Mekaaninen toteutus perustuu fontin pienentämiseen.

tin ominaisuutta. Jos käytössä oleva fontti ei sisällä haluttujen merkkien ylä- tai alaindeksiä, `realscripts`-paketin komennot käyttävät automaattisesti Latexin mekaanista keinoa. Paketti määrittelee pari muutakin hyödyllistä komentoa, muun muassa tähdelliset versiot edellä mainituista: `\textsuperscript*` ja `\textsubscript*`. Nämä komennot toteuttavat aina mekaanisen ylä- tai alaindeksin eli toimivat kuten Latexin alkupe räiset komennot.

Taulukossa 2.3 vertaillaan oikeita ja mekaanisia ylä- ja alaindeksejä. Taulukon esimerkit paljastavat, että lopputuloksessa on eroa. Mekaaninen ylä- ja alaindeksitoiminto jättää merkit turhan suurikokoisiksi mutta saattaa silti ohentaa merkkien viivoja liian paljon. Se ei myöskään ymmärrä poistaa gemenanumeroita (3624) käytöstä vaan latoo ne sellaisenaan suunnilleen oikeaan paikkaan.

Ylä- ja alaindeksejä käytettäessä on siis syytä ladata `realscripts`-paketti ja käyttää indeksit hallitsevaa Open Type -fonttia. Fonttien ominaisuuksia voi tutkia käyttöjärjestelmän komentotulkissa komennolla `otfinfo`. Toisaalta fonttiin sisältyviä ylä- ja alaindeksejä voi myös kirjoittaa Unicode-merkistön avulla sellaisenaan. Lopputulos on sama.

2.1.13 Tavutusvihje

Komento `\` on tavutusvihje, joka neuvoo rivejä latovalle algoritmille, että sanan voi katkaista tästä kohdasta rivin lopussa. Tavutusvihje voi estää sanan katkaisemisen muista kohdista.

```
tavutus\-algo\-ritmi
```

Merkki	Merkitys	Merkki	Merkitys	
à	\`a	ø	\O	poikkiviiva-O
á	\'a	ø	\o	poikkiviiva-o
â	\^a	Ð	\DJ	poikkiviiva-D
ã	\~a	đ	\dj	poikkiviiva-d
ä	\"a	Ð	\DH	versaali-eth
ǎ	\H a	ð	\dh	gemena-eth
å	\r a	Ŋ	\NG	versaali-äng
â	\v a	ŋ	\ng	gemena-äng
ă	\u a	ß	\SS	versaali kaksois-s
ã	\=a	ß	\ss	gemena kaksois-s
ǎ	\b a	Þ	\TH	versaali thorn
ȁ	\c a	þ	\th	gemena thorn
à	\.a	ı	\i	pisteetön i
ȁ	\d a	ı	\j	pisteetön j
ȁ	\k a	Æ	\AE	AE-ligatuuri
Ł	\L	æ	\ae	ae-ligatuuri
ł	\l	Œ	\OE	OE-ligatuuri
		œ	\oe	oe-ligatuuri

Taulukko 2.4: Komentoja tarkkeellisten ja muiden kirjainten kirjoittamiseen.

Myös sanassa olevat yhdysmerkit ja ajatusviivat vaikuttavat sanan tavuttamiseen. Perusteellisemmin tavutusta ja sen asetuksia käsitellään luvussa 3.6.

2.1.14 Tarkkeet ja erikoismerkit

Latexissa on useita komentoja tarkkeellisten kirjainten (á š ç ö) kirjoittamiseen sekä muille merkeille, joita ei ehkä ihan helposti saa suoraan näppäimistöltä. Komentoja on koottu taulukoihin 2.4, 2.5 ja 2.6. Taulukon tarkekomennossa on käytetty a-kirjainta esimerkkinä, mutta tarke voi liittyä muihinkin kirjaimiin. Merkit voi kirjoittaa Latex-lähdedokumenttiin myös sellaisenaan, eli näiden komentojen käyttö ei ole välttämätöntä.

Tarke- ja erikoismerkkikomentojen lisäksi on olemassa sekalaisia muita komentoja erikoisempien asioiden latomiseen. Ladontajärjestelmän logojen kirjoittamiseen on komennot `\TeX` ja `\LaTeX`. Suorakulmioita voi tehdä `\rule`-komennolla, jolle annetaan argumenteiksi ainakin kaksi mittaa: leveys ja korkeus. Myös yksi hakasulkeissa an-

TeX
 ETeX

Merkki ja komennot	Merkitys
“ <code>\textquotedblleft, ``</code>	vasen lainausmerkki
” <code>\textquotedblright, ''</code>	oikea lainausmerkki
” <code>\textquotedbl</code>	yleislainausmerkki
‘ <code>\textquoteleft, \lq, `</code>	vasen puolilainausmerkki
’ <code>\textquoteright, \rq, ’</code>	oikea puolilainausmerkki, heittomerkki
’ <code>\textquotesingle</code>	yleispuolilainausmerkki ja -heittomerkki
« <code>\guillemotleft, <<</code>	vasen kulmalainausmerkki
» <code>\guillemotright, >></code>	oikea kulmalainausmerkki
< <code>\guilsinglleft</code>	vasen kulmapuolilainausmerkki
> <code>\guilsinglright</code>	oikea kulmapuolilainausmerkki
» <code>\quotedblbase</code>	rivinalinen lainausmerkki
, <code>\quotesinglbase</code>	rivinalinen puolilainausmerkki

Taulukko 2.5: Komentoja lainausmerkkien kirjoittamiseen.

nettu valinnainen argumentti on mahdollinen. Sekin on mitta ja ilmaisee, kuinka paljon suorakulmiota nostetaan tekstin peruslinjasta. Negatiivinen mitta laskee suorakulmiota alaspäin. Latexin mittoja ja mittayksiköitä käsitellään luvussa 2.4.

```
\rule{1ex}{1ex} abc \rule{3em}{.5bp} abc \rule[1ex]{3em}{.5bp}
```

⇒ ■ abc _____ abc _____

Komento `\strut` latoo näkymättömän, leveydettömän merkin, jonka korkeus on rivikorkeuden eli mitan `\baselineskip` mukainen. Komento voi tarvita joskus esimerkiksi laatikoiden sisällä (luku 2.6).

Käytännössä `\strut`-komento hyödyntää sisäisesti `\rule`-komentoa ja latoo sen avulla leveydettömän suorakulmion. Tarkemmin sanottuna korkeus eli tekstin peruslinjan yläpuolinen osa on 0,7 kertaa `\baselineskip`-mitta ja syvyys eli peruslinjan alapuolinen osa on 0,3 kertaa `\baselineskip`.

```
\rule[-0.3\baselineskip]{0bp}{\baselineskip}
```


Merkki ja komennot	Merkitys
– \textendash, --	lyhyt ajatusviiva
— \textemdash, ---	pitkä ajatusviiva
¡ \textexclamdown, !`	ylösalainen huutomerkki
¿ \textquestiondown, ?^	ylösalainen kysymysmerkki
> \textgreater	suurempi kuin -merkki
< \textless	pienempi kuin -merkki
... \textellipsis, \ldots, \dots	kolme pistettä, ellipsi
€ \texteuro	euron merkki
£ \textsterling, \pounds	punnan merkki
\$ \textdollar, \\$	dollarin merkki
§ \textsection, \S	pykälän merkki
¶ \textparagraph, \P	kappaleen merkki
© \textcopyright, \copyright	tekijänoikeusmerkki
® \textregistered	rekisteröity tavaramerkki
™ \texttrademark	tavaramerkki
† \textdagger, \dag	risti
‡ \textdaggerdbl, \ddag	kaksoisristi
^ \textasciicircum, \^{}	sirkumfleksi
~ \textasciitilde, \~{}	tilde
* \textasteriskcentered	rivinkeskinen asteriski, tähti
\ \textbackslash	kenoviiva
\textbar	pystyviiva
\textbardbl	kaksoispystyviiva
{ \textbraceleft, \{	vasen aaltosulje
} \textbraceright, \}	oikea aaltosulje
• \textbullet	luetelmaympyrä
○ \textbigcircle	suuri ympyrä
← \textleftarrow	nuoli vasemmalle
→ \textrightarrow	nuoli oikealle
ª \textordfeminine	feminiininen järjestysluvun merkki
º \textordmasculine	maskuliininen järjestysluvun merkki
· \textperiodcentered	rivinkeskinen piste
_ \textunderscore, _	alaviiva
␣ \textvisiblespace	näkyvä välilyönti

Taulukko 2.6: Komentoja erikoismerkkien kirjoittamiseen.

2.2 Komennot

Latexin komennot alkavat kenoviivalla (`\`), jonka jälkeen tulee komennon nimi. Nimi koostuu yleensä pienistä tai isoista kirjaimista, mutta komento voi koostua myös muista merkeistä.

Komennot voivat ottaa vastaan argumentteja eli lisätietoa, jota komento käsittelee ja tarvitsee toimintaansa. Jotkin argumentit voivat olla pakollisia ja jotkin valinnaisia. Pakolliset kirjoitetaan aaltosulkeisiin `{...}` ja valinnaiset hakasulkeisiin `[...]`.

```
\komento
\komento{argu}{mentteja}
\komento[valinnainen]{argu}{mentteja}
```

Jos pakolliseen argumenttiin haluaa sisällyttää aaltosulkeen, täytyy sen eteen kirjoittaa kenoviiva `\{ \}`, tai voi myös käyttää taulukossa 2.6 mainittuja komentoja aaltosulkeiden tuottamiseen. Sama pätee aaltosulkeiden latomiseen muutenkin.

Hakasulkeet sen sijaan ladotaan tekstiin normaalisti, eikä niiden kanssa käytetä kenoviivaa. Poikkeustilanne on komennon valinnaisen argumentin sisällä, koska valinnainen argumentti jo sinänsä kirjoitetaan hakasulkeiden sisään. Hakasulkeita ei voi suojata kenoviivalla, koska `\[ja \]` ovat jo muuhun tarkoitettuja komentoja: niillä luodaan matematiikkatilassa (luku 4.16) oleva tekstilohko. Valinnaisen argumentin sisään saa hakasulkeen, kun sen kirjoittaa aaltosulkeiden sisään. Esimerkiksi komennon `\komento[{}]` valinnaiseksi argumentiksi tulee lopulta yksi `]`-merkki.

Komennon yhteydessä sanavälejä käsitellään hieman poikkeuksellisesti. Esimerkiksi komennon nimen perässä olevat sanavälit syödään pois, jos komennolle ei anneta yhtään argumenttia. Seuraavassa esimerkissä sana ”`Latex`” ladotaan ehjänä, jos vain `\komento` itsessään ei kirjoita mitään eikä vaikuta tekstin latomiseen.

```
La\komento tex
⇒ Latex
```

Jos täytyy saada komennon nimen jälkeinen sanaväli näkyviin, täytyy kirjoittaa komennon nimen perään aaltosulkeet (`\komento{}`) tai kenoviiva (`\komento\`).

Komennon nimen ja argumenttien välissä voi olla sanavälejä, ja ne kaikki syödään pois. Komennon ja sen argumentit voi siis kirjoittaa vaikka seuraavalla tavalla:

```
\komento [valinnainen]
  {argu-} {mentteja}
```

2.2.1 Omat komennot ja abstrahointi

Omien komentojen tärkein tarkoitus on merkinäytösten abstrahointi eli teknisen toteutuksen ja yksityiskohtien piilottaminen. Sopiva abstrahointi helpottaa lähdedokumentin käsittelemistä.

Esimerkiksi jos kirjoittaa Latexia käsittelevää kirjaa, kannattaa heti aluksi luoda komento, jolla merkitään kaikki Latexin komennot. Komennon nimi voisi olla vaikka `\komento`, ja sen voisi määritellä siten, että se lisää automaattisesti komennon nimen alkuun kenoviivan (`\`) ja latoo koko ilmauksen tasalevyisellä fontilla. Seuraavassa on esimerkki tällaisen komennon määrittämisestä ja käytöstä:

```
\newcommand{\komento}[1]{\texttt{\textbackslash #1}}
```

```
Komennolla \komento{section} tehdään otsikoita.
```

⇒ Komennolla `\section` tehdään otsikoita.

Edellä mainitun komennon määritelmään voi olla tarpeen lisätä myöhemmin muitakin asioita. Latexin komentojen nimet ovat suunnilleen englantia, ja jos nimen tavuttaa, täytyisi se tehdä englannin sääntöjen mukaisesti. Esimerkkinä olleen `\komento`-komennon määritelmässä voisi siis vaihtaa myös kielen komennolla `\textenglish`.⁴

Jos ollaan kirjoittamassa laajaa tietoteosta, Latex-komennot halutaan ehkä lisätä automaattisesti kirjan lopussa olevaan asiahakemistoon (ks. s. 257). Niinpä komennon määritelmään lisätään vielä sitäkin varten komento `\index`.⁵ Lopulta Latex-komentojen merkitsemiseen tarkoitettu `\komento` määriteltäisiin seuraavalla tavalla:

```
\newcommand{\komento}[1]{%
  \texttt{\textbackslash\textenglish{#1}}%
  \index[komennot]{#1@\texttt{\textbackslash #1}}
```

⁴ Kieliasetuksista käsitellään luvussa 3.5.

⁵ Asiahakemistoja käsitellään luvussa 4.14.

Näin oma `\komento` ilmaisee tiiviisti ja havainnollisesti tarkoituksen eli sen, että kyseessä on Latex-komento. Se piilottaa monimutkaisen teknisen toteutuksen eli fontin ja kielen vaihtamiseen sekä asiahakemistoon liittyvät toiminnot. Lisäksi komennon teknistä toteutusta on helppoa muuttaa myöhemmin, koska komennon määrittely on vain yhdessä paikassa lähdedokumentin alussa.

2.2.2 Komentojen määrittely

Komentojen määrittelyyn on kolme erilaista komentoa, ja niille kaikille annetaan samanlaiset argumentit. Komennot ovat seuraavat:

```
\newcommand    {\nimi}[n][oletus]{määritelmä}
\renewcommand  {\nimi}[n][oletus]{määritelmä}
\providecommand{\nimi}[n][oletus]{määritelmä}
```

Ensimmäinen pakollinen argumentti on komennon nimi (`\nimi`), ja se voi koostua vain kirjaimista. Komento `\newcommand` määrittelee uuden komennon. Mikäli komento on jo olemassa, annetaan virheilmoitus. Toinen komento `\renewcommand` määrittelee olemassa olevan komennon uudelleen. Se antaa virheilmoituksen, jos komentoa ei ollut olemassa. Kolmas komento `\providecommand` puolestaan määrittelee uuden komennon vain siinä tapauksessa, että sellaista ei ollut ennen olemassa. Se ei anna virheilmoitusta.

Komentojen toinen pakollinen argumentti määritelmä sisältää komennon määritelmän eli mitä tahansa tekstiä tai komentoja. Suoritusvaiheessa komento ikään kuin vaihdetaan sen määritelmäksi.

```
\newcommand{\komento}{Minua komennettiin!}
\komento
```

⇒ Minua komennettiin!

Komentojen ensimmäinen valinnainen argumentti `n` on luku, joka kertoo, kuinka monta argumenttia määriteltävä komento käsittelee. Määritelmässä voi käyttää parametreja #1, #2, #3 jne., ja ne korvautuvat komennon suoritusvaiheessa ensimmäisellä, toisella, kolmannella jne. argumentilla.

```
\newcommand{\komento}[2]{Sanoit #1 ja #2!}
\komento{hip}{hei}
```

⇒ Sanoit hip ja hei!

Toinen valinnainen argumentti oletus – jos se on mukana – kertoo, että määriteltävän komennon ensimmäinen argumentti on valinnainen ja että tämä on sen oletusarvo. Oletusarvoa käytetään silloin, kun valinnaista argumenttia ei ole annettu.

```
\newcommand{\komento}[3][tyyppi]{Hei #1, sanoit #2 ja #3!}  
\komento{hip}{hei} \\  
\komento[Leslie]{hip}{hei}
```

⇒ Hei tyyppi, sanoit hip ja hei!
Hei Leslie, sanoit hip ja hei!

Joskus yhden komennon määritelmä sisältää komennon `\renewcommand`, joka sitten määrittelee uudelleen jonkin toisen komennon. Silloin parametrit #1, #2 jne. on tarkoitettu ensimmäisen eli uloimman kerroksen käsiteltäväksi. Sisempi kerros käyttää parametreja ##1, ##2 jne.

Kaikista kolmesta komentojen määrittelykomennosta on olemassa tähdellinen versio eli sellainen, jonka komennon nimen lopussa on tähti (*). Latexin komennoissa on tapana, että tähdellinen versio – jos sellainen on olemassa – tarjoaa samaan asiaan jonkin toisenlaisen näkökulman.

```
\newcommand*   {\nimi}[n][oletus]{määritelmä}  
\renewcommand* {\nimi}[n][oletus]{määritelmä}  
\providecommand* {\nimi}[n][oletus]{määritelmä}
```

Komentojen määrittelyssä tähdelliset versiot toimivat muuten samalla tavalla, mutta ne antavat virheilmoituksen, jos komennolle annetut argumentit sisältävät enemmän kuin yhden tekstikappaleen. Niinpä esimerkin 2.1 koodi tuottaa käännettäessä virheen.

`\newcommand*`-komennolla määritelty `\komento` ei siis suostu ottamaan vastaan argumentteja, jotka sisältävät kappaleen vaihtumisen eli enemmän kuin yhden tekstikappaleen. Tämä voi olla hyödyllinen suojausominaisuus.

2.2.3 Kestävät ja hauraat komennot

Kaikki Latex-komennot ovat joko kestäviä (robust) tai hauraita (fragile). Tällä ei ole yleensä käytännön merkitystä, mutta hauraat komennot eivät välttämättä toimi toisen komennon argumentissa.

```

1 \newcommand*{\komento}[1]{Teksti: #1}
2
3 \komento{
4   Ensimmäinen tekstikappale.
5
6   Toinen tekstikappale.
7 }

```

Esimerkki 2.1: `\newcommand*`-komennolla määritelty komento ei salli argumentteja, joissa on useita tekstikappaleita. Tämä esimerkki tuottaa käännettäessä virheilmoituksen.

Hauraita komentoja ovat sellaiset, jotka sisältävät tietoa, jota Latex kirjoittaa väliaikaistiedostoon ja lukee sieltä takaisin. Kyse on esimerkiksi sisällysluetteloon tai muihin automaattisiin luetteloihin kirjoitetavasta tiedosta. Hauraita ovat myös rivinvaihdot ja valinnaisia argumentteja eli hakasulkeissa annettavia argumentteja sisältävät komennot.

Käytännössä ongelmia aiheuttavat esimerkiksi otsikkokomennot (`\section` ym.) sekä kuvatekstikomento `\caption`. Näiden argumenttina oleva teksti (ja mahdolliset komennot) kirjoitetaan tiedostoon ja ladataan sieltä myöhemmin takaisin. Hauraat komennot eivät toimi edellä mainittujen komentojen argumentissa, ja lähdetiedoston kääntäminen johtaa virheilmoitukseen. Ongelman voi ainakin joskus korjata kirjoittamalla hauraan komennon eteen komennon `\protect`.

2.2.4 Muita vinkkejä

Komennon viimeistä argumenttia ei välttämättä tarvitse kirjoittaa aaltosulkeisiin, jos argumentiksi halutaan vain yksi merkki. Tällaisessa tilanteessa komento poimii argumentiksi seuraavan merkin, joka ei ole sanaväli.

```

\newcommand{\x}[1]{Argumentti: <#1>}
\x abc \
\x.abc

```

⇒ Argumentti: <a>bc
Argumentti: <.>abc

```

1 \newcommand{\komento}{alkuperäinen}
2 \komento
3 {%
4   \renewcommand{\komento}{\textit{muutettu}}
5   \komento
6 }
7 \komento

```

⇒ alkuperäinen *muutettu* alkuperäinen

Esimerkki 2.2: Aaltosulkeilla voi rajata komennon määrittelyn vaikutusalueita.

Mikäli argumenttina on kenoviivalla alkava komento, sitäkään ei tarvitse kirjoittaa aaltosulkeisiin. Latex-koodin lukemisen kannalta tällainen ei välttämättä ole hyvä käytäntö, koska joskus voi hämärtyä, onko kyse kahdesta peräkkäisestä komennosta vai onko toinen komento vain argumenttina toiselle.

```

\newcommand{\x}[1]{Argumentti: <#1>}
\newcommand{\yyy}{abc}
\x\yyy

```

⇒ Argumentti: <abc>

Tätä merkintätapaa esiintyy jokin verran komentojen määrittelyssä, niin että jätetään `\newcommand`-komennon ensimmäisenä argumenttina oleva komennon nimi ilman aaltosulkeita.

```

\newcommand\yyy{abc}

```

Komennon määrittelyssä on välillä hyötyä `\ignorespaces`-komennosta, joka jättää sanavälit huomioimatta komennon jälkeen. Ilman tätä komentoa tulisi seuraavassa esimerkissä sanojen väliin yksi välilyönti.

```

\newcommand{\komento}[1]{#1\ignorespaces}
\komento{yhdyss} sana

```

⇒ yhdyssana

Aaltosulkeilla (luku 2.1.6) voi rajata komentomäärittelyn vaikutusalueita. Esimerkin 2.2 alussa asetetaan `\komento` tiettyyn alkuperäismäärittelmään. Aaltosulkeiden sisällä se määritellään väliaikaisesti uudestaan.

Aaltosulkeilla rajatun ympäristön jälkeen komennon uusi määritelmä lakkaa ja komento palautuu alkuperäiseksi.

Monimutkaisiin komentoihin voidaan tarvita ehtorakenteita. Ne saa toteutettua `ifthen`⁶-paketin tarjoaman `ifthenelse`-komennon avulla. Se on ohjelmointikielistä tuttu ehtorakenne: jos annettu ehtolauseke on tosi, käsitellään `then`-haara; muussa tapauksessa käsitellään `else`-haara.

2.3 Ympäristöt

Ympäristöt ovat rakenteita, joilla on aloittava `\begin`-komento ja lopettava `\end`-komento sekä nimi. Ympäristöjen ajatuksena on, että jokin ominaisuus tai jotkin toiminnot ovat voimassa vain ympäristön sisällä ja asiat palautuvat ennalleen ympäristön jälkeen. Tässä mielessä ne toimivat samalla tavalla kuin aaltosulkeet (luku 2.1.6). Jos esimerkiksi fonttiasetusta (luku 3.4) muuttaa ympäristön sisäpuolella, asetus palautuu ympäristön jälkeen samaksi kuin se oli ennen ympäristön alkua. Samoin ympäristön sisällä määritellyt komennot ovat voimassa vain kyseisessä ympäristössä.

```
\begin{nimi}
% ympäristön
% vaikutusalue
\end{nimi}
```

Yleisin ympäristö on nimeltään `document`, jonka sisään koko dokumentin sisältö kirjoitetaan. Muita ympäristöjä käytetään tavallisesta leipätekstistä poikkeavien rakenteiden ilmaismiseen, esimerkiksi luettelmiin ja taulukoihin (luvut 4.6 ja 4.7). Ympäristöjä voi tehdä itsekin mihin hyvänsä tarkoitukseen. Niitä määritellään seuraavilla komenoilla:

```
\newenvironment {nimi}[n][oletus]{aloitus}{lopetus}
\renewenvironment {nimi}[n][oletus]{aloitus}{lopetus}
```

Komento `\newenvironment` määrittelee uuden ympäristön. Se antaa virheilmoituksen, jos samanniminen ympäristö on jo olemassa. Komento

⁶ <https://www.ctan.org/pkg/ifthen>

`\renewenvironment` puolestaan määrittelee uudelleen ympäristön, joka on jo olemassa. Se antaa virheilmoituksen, jos ympäristöä ei ollutkaan olemassa.

Argumentit ovat lähes samanlaiset kuin komentojen määrittelyssä (luku 2.2.2). Ympäristöjen määrittelykomennoilla on kolme pakollista argumenttia: ensimmäinen on ympäristön nimi, toinen on ympäristön aloitusmääritelmä (aloitus) ja kolmas on lopetusmääritelmä (lopetus).

```
\newenvironment{ymp}{Tästä se alkaa.}{Tähän se päättyy.}

\begin{ymp}
  Ympäristön sisältöä.
\end{ymp}
```

⇒ Tästä se alkaa. Ympäristön sisältöä. Tähän se päättyy.

Tavallisesti ympäristön aloitusmääritelmään kirjoitetaan jonkin toisen ympäristön aloituskomento sekä mahdollisesti suoritetaan joitakin asetuskomentoja. Vastaavasti lopetusmääritelmässä lopetetaan ympäristö eli palataan normaaliin tilaan. Tarkoituksena on abstrahoida jokin monimutkaisempi kokonaisuus eli tehdä uusi helppokäyttöinen ympäristö, joka piilottaa teknisen toteutuksen.

```
\newenvironment{ymp}
{\begin{mahtavuus}
  \omia\hienoja\asetuksia}
{\end{mahtavuus}}
```

Omille ympäristölle voi määritellä argumentteja samalla tavalla kuin komennoillekin. Ensimmäinen valinnainen argumentti n on luku joka kertoo, kuinka monta argumenttia määriteltävä ympäristö käsittelee. Ympäristön aloitusmääritelmässä voi argumentteihin viitata parametreilla #1, #2, #3 jne.

Jos toinen valinnainen argumentti oletus on mukana, se ilmaisee, että määriteltävän ympäristön ensimmäinen argumentti on valinnainen ja että tämä on sen oletusarvo. Oletusta käytetään silloin, kun valinnaisista argumentteista ei ole annettu. Argumentit annetaan ympäristön aloitustavan `\begin`-komennon yhteydessä.

```

1 \newenvironment{ymp}
2   {Yhdys\ignorespaces}
3   {esi\ignorespacesafterend}
4
5 \begin{ymp}
6   sana% Kommentti poistaa seuraavan sanavälin.
7 \end{ymp}   merkki.

```

⇒ Yhdyssanaesimerkki.

Esimerkki 2.3: Sanavälien käyttäytyminen ympäristöjen yhteydessä. Komennoilla `\ignorespaces` ja `\ignorespacesafterend` voi poistaa seuraavat sanavälit.

```

\begin{ymp}[valinnainen]{argu}{mentteja}
% ympäristön
% vaikutusalue
\end{ymp}

```

Ympäristön määrittelykomennoille on myös tähdelliset versiot `\newenvironment*` ja `\renewenvironment*`. Ne toimivat samoin kuin edellä kuvatut tavallisetkin komentoversiot, mutta ne eivät salli, että määritellylle ympäristölle annetut argumentit sisältävät enemmän kuin yhden tekstikappaleen. Toiminta on siis sama kuin komentojenkin määrittelyyn tähdellisissä versioissa (luku 2.2.2).

```

\newenvironment* {nimi}[n][oletus]{aloitus}{lopetus}
\renewenvironment* {nimi}[n][oletus]{aloitus}{lopetus}

```

Joskus ympäristöjen määrittelyyn on hyödyllistä sisällyttää komento `\ignorespaces`, joka jättää huomioimatta tämän komennon jälkeiset sanavälit. Toinen hyödyllinen on `\ignorespacesafterend`, joka jättää huomioimatta ympäristön lopettavan `\end`-komennon jälkeiset sanavälit. Esimerkki 2.3 selventää näiden toimintaa.

2.4 Mitat

2.4.1 Mittayksiköt

Koska typografia on pitkälti teksti- ja muiden elementtien sijoittelua, tarvitaan sitä varten mittavälineitä. Niinpä Texissäkin on pituusmitto-

Lyh.	Merkitys
bp	piste uudessa pica-järjestelmässä, 1/72 tuumaa, 0,3528 mm
pt	piste vanhassa pica-järjestelmässä, 1/72,27 tuumaa, 0,3515 mm
pc	pica eli 12 pt-pistettä
sp	1/65536 pt-pistettä (5,36 nm), Texin sisäisesti käyttämä yksikkö
dd	piste Didot-järjestelmässä, 0,376 mm
cc	cicero eli 12 dd-pistettä
mm	millimetri
cm	senttimetri
in	tuumaa, 25,4 mm
ex	nykyisen fontin x-korkeus, perus- ja gemenalinjan etäisyys
em	typografisen neliön sivun pituus, sama kuin fontin koko

Taulukko 2.7: Texin mittayksiköiden lyhenteet ja merkitykset.

ja ja useita pituuden mittayksiköitä. Taulukkoon 2.7 on koottu mittayksiköiden lyhenteet ja merkitykset. Teknisesti on samantekevää, mitä yksiköitä käyttää, sillä ne ovat vain välineitä pituuden ilmaisemiseen. Sisäisesti Tex käyttää sp-yksikköä, joka ilmaisee samalla mittojen tarkkuuden: pienin jakamaton mitta on 1 sp (5,36 nm).

Usein tietyt yksiköt ovat vakiintuneet tiettyihin tilanteisiin. Esimerkiksi fonttien kokoja ja rivikorkeuksia on tapana mitata pistemittojen avulla. Nykyään käytetään lähinnä bp-yksikön mukaista pistettä, joka tuli käyttöön Post Script -standardin myötä vuonna 1984 ja jota käytetään julkaisuohjelmissa. Sivun mittoja kuten leveyttä, korkeutta ja marginaaleja ilmaistaan tavallisesti metrijärjestelmän avulla eli senttimetreissä tai millimetreissä.

Latexin mittojen mittaluvuissa desimaalierottimena on piste, ja mittayksikön lyhenne kirjoitetaan kiinni mittalukuun. Seuraavassa esimerkissä tehdään vaakasuuntaisia ja pystysuuntaisia välejä komennoilla `\hspace` ja `\vspace`:

```
Sanat \hspace{1.2cm}hassusti
\vspace{2mm}
```

```
\hspace{1.75em}erillään.
```

```
⇒ Sanat      hassusti
           erillään.
```

Jos pystysuuntaisen välin tekevä komento `\vspace` sattuu sivunvaihdon kohdalle, väli jätetään kokonaan tekemättä. Komennosta on olemassa tähtiversio `\vspace*`, joka tekee välin myös sivunvaihdon kohdalle eli ennen tai jälkeen sivunvaihdon.

Vastaavalla tavalla toimii myös vaakasuuntainen väli `\hspace`. Jos komento sattuu rivinvaihdon kohdalle, komentoa ei huomioida eli väli jätetään tekemättä. Tähtiversio `\hspace*` tekee välin joka tapauksessa.

2.4.2 Mittakomennot ja typografinen viivasto

Mittoja tallennetaan eräänlaisiin muuttujiin, jotka näyttävät päällepäin komennoilta eli niiden alussa on kenoviiva (`\`) ja sitten kirjaimista koostuva nimi. Esimerkiksi mitta `\textwidth` on tekstialueen leveys nykyisellä sivulla. Komentomaisesta ulkoasustaan huolimatta mittoja ei voi suorittaa komentoina; ne sopivat vain komennon argumentiksi, silloin kun tarvitaan mitta.

Uusia mittoja luodaan komennolla `\newlength` ja olemassa olevia mittoja asetetaan esimerkiksi komennolla `\setlength` ja `\addtolength`. Omien mittoja on tarpeen luoda silloin, kun halutaan määritellä tietynsuuruinen mitta, jota käytetään Latex-koodissa useita kertoja.

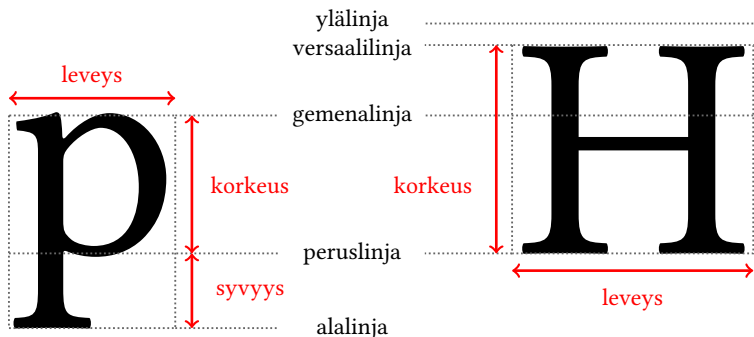
```
\newlength{\omamitta}           % Luodaan mitta.
\setlength{\omamitta}{2.3em}    % Asetetaan mitta.
\addtolength{\omamitta}{1em}   % Lisätään mittaan.
\addtolength{\omamitta}{-1em}  % Vähennetään mitasta.
```

Näin luotuja mittoja voi käyttää mittayksiköiden tavoin, eli niille voi asettaa eteen kertoimen. Esimerkiksi seuraava `\hspace`-komento luo vaakasuuntaisen välin, jonka pituus on 0,7 kertaa `\omamitta`:

```
\hspace{0.7\omamitta}
```

Koska vaakasuuntaisten välien tekeminen on typografiassa varsin tavallista, on niitä varten olemassa omia komentoja. Typografisen neliön levyisen (1 em) välin voi tehdä komennolla `\quad`. Sen puolikkaan ($\frac{1}{2}$ em) saa komennolla `\enspace`. Ohuke eli $\frac{1}{4}$ em-väli tehdään `\,`-komentolla, josta on tarkempaa tietoa luvussa 2.1.8.

Latex sisältää myös komennot mittojen poimimiseen kirjaimista tai muista merkeistä. Mittoja on kolme: leveys (width), korkeus (height)



Kuva 2.1: Kirjainten mitat ja typografinen viivasto.

ja syvyys (depth). Leveys on merkin viemä tila leveyssuunnassa. Korkeus tarkoittaa merkin korkeutta peruslinjan yläpuolella, ja syvyys on merkin korkeus peruslinjan alapuolella. Kuva 2.1 havainnollistaa näitä kolmea mitta sekä typografista viivastoa.

Gemenalinja sijaitsee gemenakirjainten eli pienaakkosten korkeudella. Peruslinjan ja gemenalinjan etäisyys on fontin x-korkeus (1 ex). Optisen vaikutelman vuoksi gemenalinja ei tarkoita ehdotonta gemenakirjainten ylintä kohtaa. Esimerkiksi p-kirjaimen yläpääte ja silmukan yläkaari voivat yltää hieman gemenalinjan yläpuolelle ja silmukan alakaari peruslinjan alapuolelle. Fonttien suunnittelijat tekevät tällaisia optisia korjauksia, jotta kirjainten hahmot näyttäivät yhtä korkeilta ja tasapainoisilta yhdessä.

Versaalilinja sijaitsee versaalikirjainten eli suuraakkosten korkeudella. Jotkin kirjaimet voivat yltää hieman versaalilinnan yläpuolellekin. Esimerkiksi joidenkin fonttien Hk-kirjaimia vertailemalla näkee, että k-kirjaimen yläpääte yltää H-kirjaimen eli versaalilinnan yläpuolelle, ylälinjan tuntumaan. Myös tarkemerkit (luku 2.1.14) voivat sijaita versaalilinnan yläpuolella.

Merkkien leveyden, korkeuden ja syvyyden voi mitata seuraavan esimerkin komennoilla. Esimerkin ensimmäisellä rivillä luodaan uudet mitat `\leveys`, `\korkeus` ja `\syvyys`, joihin sitten tallennetaan merkkien mitat.

```

\newlength{\leveys} \newlength{\korkeus} \newlength{\syvyys}
\settowidth{\leveys}{abc} % merkkien "abc" leveys
\settoheight{\korkeus}{H} % merkin "H" korkeus
\settoheight{\syvyys}{p} % merkin "p" syvyys

```

Mitat saa näkyviin kirjoittamalla mitan eteen komennon `\the`. Näin mitan pituus ladotaan dokumenttiin. Yksikkönä on typografinen piste (pt).

```
Leveys: \the\leveys, korkeus: \the\korkeus, syvyys: \the\syvyys.
```

⇒ Leveys: 14.62865pt, korkeus: 6.81897pt, syvyys: 2.4662pt.

Aiemmin mainituista optisista korjauksista johtuu, että gemenakirjaimet eivät ole välttämättä täsmälleen samankorkuisia. Gemena-x asetuu perus- ja gemenalinjojen väliin, mutta gemena-o voi yltää aavistuksen verran linjojen ylä- ja alapuolelle. Ihmisen silmään ne näyttävät yhtä korkeilta.

2.4.3 Venyvät mitat ja välit

Edellä on puhuttu vain kiinteistä mitoista, mutta Latex tuntee myös venyvät mitat. Niiden ajatuksena on, että Latexille voi antaa luvan kutistaa tai venyttää mittaa tietyissä rajoissa. Venyviä mittoja käytetään usein pystysuuntaisissa väleissä, esimerkiksi väliotsikon edellä ja jälkeen tai tekstikappaleiden välissä. Latex pystyy latomaan sivut yleensä paremman näköiseksi, kun sille antaa venyvien mittojen avulla hieman säätövaraa. Alla on esimerkki venyvän kappalevälin (`\parskip`) määrittämisestä. Samalla asetetaan kappaleen ensimmäisen rivin sisennys (`\parindent`) nolnaan (0em).

```

\setlength{\parskip}{2ex plus 0.2ex minus 0.1ex}
\setlength{\parindent}{0em}

```

Venyvissä mitoissa mainitaan ensin normaali pituus ja sitten sanoilla plus ja minus ilmaistaan, kuinka paljon mitta voi venyä tai kutistua. Molempia ei välttämättä tarvitse antaa.

Venyvät mitat voivat sisältää ”äärettömän” mittayksikön fill, joka antaa luvan venyttää mittaa niin, että kaikki käytettävissä oleva tila täyttyy. Äärettömän mittayksikön kanssa mittaluku ilmaisee suhdeluvun muihin äärettömiin mittoihin. Seuraava esimerkki havainnollistaa asiaa:

```
x\hspace{0mm plus 1fill}x\hspace{0mm plus 2fill}x
```

⇒ x x x

Edellisen esimerkin mittojen luonnollinen pituus on nolla (0mm), mutta plus ja fill -mitan vuoksi ne voivat venyä ja täyttää koko käytettävissä olevan tilan. Ensinnä mainitulla on suhdeluku 1 (1fill) ja jälkimmäisellä suhdeluku 2 (2fill), joten jälkimmäinen väli on kaksinkertainen ensimmäiseen verrattuna. Saman äärettömästi venyvän mitan ja suhdeluvut voi ilmaista myös `\stretch`-mittakomennon avulla seuraavasti:

```
x\hspace{\stretch{1}}x\hspace{\stretch{2}}x
```

⇒ x x x

Äärettömästi venyvälle, koko tilan täyttävälle mittayksikölle on itse asiassa kolme eri versiota: `fil`, `fill` ja `filll`. Niiden erona on tärkeysjärjestys. Ensinnä mainittu `fil` on vähiten tärkeä, ja viimeinen eli `filll` on tärkein. Jos samassa yhteydessä käytetään eri tärkeysasteisia yksiköitä, ylempiantasoiset mitätöivät alemmantasoiset.

```
x\hspace{0mm plus 1filll}x\hspace{0mm plus 1fill}x
```

⇒ x xx

Edellisessä esimerkissä ensimmäinen venyvä mitta `1filll` mitätöi jälkimmäisen `1fill` kokonaan. Yleensä lienee parasta käyttää keskimäistä (`fill`), mutta eri tärkeysasteille voi olla käyttöä esimerkiksi laajennuspaketin koodissa: `fil` sallii, että kirjoittaja tai muu koodi syrjäyttää asetuksen; `filll` on ehdoton sääntö, joka syrjäyttää muut.

Käytännöllisyyssyistä äärettömästi venyvä pituus `0mm plus 1fill` on jo valmiiksi asetettu mittaan `\fill`. Sitä on kätevää käyttää esimerkiksi vaak- ja pystysuuntaisia välejä latovien kommentojen `\hspace` ja `\vspace` kanssa:

```
\hspace{\fill} % ääretön vaakasuuntainen väli  
\vspace{\fill} % ääretön pystysuuntainen väli
```

Edellisen esimerkin komennoista on olemassa vieläkin lyhemmät versiot. Vaakasuuntainen ääretön väli syntyy myös komennolla `\hfill` ja pystysuuntainen väli komennolla `\vfill`. Seuraavassa toistetaan eräs aiempi esimerkki yksinkertaisemmalla tavalla:

part	paragraph	figure	enumi
chapter	subparagraph	table	enumii
section	page	footnote	enumiii
subsection	equation	mpfootnote	enumiv
subsubsection			

Taulukko 2.8: Latexin laskurit.

```
x\hfill x\hfill\hfill x
⇒ x                                     x
```

2.5 Laskurit

Laskureiden avulla Latex pitää kirjaa esimerkiksi sivunumeroista, lukujen ja alalukujen sekä kuvien ja muiden elementtien numeroinnista. Esimerkiksi nyt olemme pääluvun 2 alaluvussa 5, ja mainitut luvut tulevat Latexin laskureista automaattisesti.

Taulukossa 2.8 ovat Latexin perus laskurit. Monet niistä ovat otsikoiden numerointia varten (`part`, `chapter`, `section`, `paragraph`). Sivunumeroita varten on `page`-laskuri ja matemaattisten kaavojen numerointiin `equation`-laskuri. Leijuvista kuvista ja taulukoista (luku 4.9) pidetään kirjaa laskureissa `figure` ja `table`, ja alaviitteiden (luku 4.11) numerointi on laskureissa `footnote` ja `mpfootnote`. Numeroidut luettelmat (luku 4.6) käsitellään `enum`-alkuisten laskureiden avulla siten, että perustasolla käytetään laskuria `enumi`, ja jos sen luettelmakehoita sisältää toisen numeroidun luettelman, käytetään siinä laskuria `enumii`, sen sisällä laskuria `enumiii` jne.

Edellä mainituista laskureista ei tarvitse yleensä itse huolehtia, sillä ne ovat vain tekniikkaa, joka toimii korkeamman tason toimintojen taustalla. Joskus on kuitenkin käyttöä myös omille laskureille. Seuraava esimerkki esittelee laskureiden käsittelyn peruskomennot. Niissä käytetään laskuria nimeltä `oma`. Laskurien nimet koostuvat pelkistä kirjaimista, eikä niiden alussa ole kenoviivaa niin kuin kommentojen ja mittojen alussa.

Komento	Merkitys
<code>\arabic</code>	arabialainen luku: 1, 2, 3...
<code>\roman</code>	roomalainen luku: i, ii, iii...
<code>\Roman</code>	roomalainen luku: I, II, III...
<code>\alph</code>	kirjain: a, b, c... (vain 1–26)
<code>\Alph</code>	kirjain: A, B, C... (vain 1–26)
<code>\fnsymbol</code>	symboli: *†‡§¶... (vain 1–9)

Taulukko 2.9: Komennot laskurien arvon latomiseen.

```

\newcounter{oma}           % Luodaan uusi laskuri "oma".
\setcounter{oma}{3}       % Asetetaan laskurin arvoksi 3.
\addtocounter{oma}{1}    % Lisätään laskurin arvoon 1.
\addtocounter{oma}{-1}   % Vähennetään laskurin arvosta 1.

```

Laskuri on sisäisesti kokonaisluku, mutta sen arvon voi latioa eri muodoissa: arabialaisena tai roomalaisena lukuna, kirjaimena tai symbolien sarjana. Taulukossa 2.9 ovat komennot laskurin arvon latomiseen. Komennon argumentiksi annetaan laskurin nimi. Laskurin voi latioa kirjainmuodossa, jos sen arvo on 1–26; symbolit toimivat vain lukualueella 1–9.

Taulukon 2.9 komentojen lisäksi laskuriin liittyvän arvon voi latioa komennolla, joka alkaa kirjaimilla `\the` ja jatkuu laskurin nimellä. Esimerkiksi komento `\thepage` latioa sivunumeron eli tekee käytännössä saman asian kuin komento `\arabic{page}`. Aina nämä eivät kuitenkaan ole sama asia, ja `\the`-alkuinen komento voi olla määritelty toisellakin tavalla. Esimerkiksi tämän oppaan leijuvat taulukot numeroidaan laskurilla `table`, mutta komento `\thetable` latioa ensin pääluvun ja sen perään pisteen ja taulukon numeron. Viimeisin taulukko on numeroltaan 2.9 eli pääluvun 2 taulukko 9.

Jos laskurin arvoa tarvitaan Latexin teknisessä tilanteessa eikä tarkoituksena ole latioa sitä näkyviin itse dokumenttiin, täytyy käyttää komentoa `\value`. Seuraava esimerkki luo laskurin nimeltä `mitta`, jonka arvoa (3) käytetään `\hspace`-komennon argumenttina mitan ilmaisemiseen (3 em).

```

\newcounter{mitta}
\setcounter{mitta}{3}
x\hspace{\value{mitta}em}x

```

⇒ x x

2.5.1 Hierarkkiset laskurit

Laskurit voivat olla hierarkkisia ja riippuvaisia toisista laskureista: kun yhden laskurin arvo kasvaa, nollautuu jokin toinen laskuri automaattisesti. Tällaista toimintoa käytetään dokumentin lukujen eli otsikoiden numeroinnissa. Kirjan pääluvun 1 alaluvut ovat esimerkiksi 1.1, 1.2, 1.3 jne. Kun alkaa seuraava pääluku 2, nollautuu alaluvun laskuri, jotta saadaan oikein numeroidut alaluvut 2.1, 2.2, 2.3 jne. Latex huolehtii lukujen numeroinnista automaattisesti, mutta omille laskureille täytyy määrittellä riippuvuussuhteet itse. Seuraavassa esimerkissä luodaan oma laskuri, joka nollautuu automaattisesti aina, kun sivu vaihtuu eli laskurin `page` arvo kasvaa.

```
\newcounter{oma}[page]
```

Ylemmän tason laskureita ei kannata kasvattaa komennolla `\addtocounter`, koska se vain muuttaa laskurin arvoa mutta ei huolehdi alemman tason laskurien nollaamisesta. Hierarkkisten laskurien kasvattamiseen on seuraavat kaksi komentoa:

```
\stepcounter{laskuri}  
\refstepcounter{laskuri}
```

Edellä mainitut komennot kasvattavat argumenttina annetun laskurin arvoa yhdellä ja nollaavat siitä riippuvaiset alemman tason laskurit. Jälkimmäinen komento `\refstepcounter` asettaa lisäksi ristiviitteen numeron, joten tämän komennon jälkeen mahdollisesti tuleva `\label`-komento luo ristiviitekohteen. Kun tähän kohtaan viitataan muualta käyttämällä `\ref`-komentoa, viitteessä näkyy laskurin numero tai oikeastaan `\thelaskuri`-komennon latoma teksti. Ristiviitteitä käsitellään tarkemmin luvussa 4.10.

Paketti `chngcntr`⁷ helpottaa hierarkkisten laskurien muuttamista jälkikäteen. Paketin komentojen avulla voi jälkikäteen asettaa jonkin laskurin riippuvaiseksi toisesta laskurista tai poistaa riippuvuuden.

⁷ <https://www.ctan.org/pkg/chngcntr>

```

\newcounter{oma}           % Luodaan laskuri "oma".
\counterwithin {oma}{page} % Asetetaan riippuvuus page-laskurista.
\counterwithout{oma}{page} % Poistetaan riippuvuus.

```

Edellä mainitut `chngcntr`-paketin komennot määrittelevät uudelleen myös `\theoma`-komennon, jolla `oma`-laskurin arvon voi latoa. Se latio mukaan myös ylemmäntasoisen laskurin arvon. Jos tätä `\the`-alkuisia komentoa ei halua määritellä uudelleen, täytyy käyttää tähdellisiä komennon versioita: `\counterwithin*` ja `\counterwithout*`. Laskuriin liittyvän `\the`-komennon voi aina itsekin määritellä haluamansa laiseksi komennolla `\renewcommand` (luku 2.2.2), esimerkiksi seuraavalla tavalla:

```

\renewcommand{\theoma}{\arabic{page}/\alph{oma}}

```

2.5.2 Kokonaislaskurit

Latex ei tiedä etukäteen, mihin arvoon laskurit lopulta yltävät. Se vain latio sivuja peräkkäin eikä tiedä mitään tulevasta. Sen vuoksi esimerkiksi dokumentin sivumäärää ei voi ihan yksinkertaisella komennolla latio dokumentin alkusivuille.

Tekninen vastaus tämäntyyppisiin ongelmiin löytyy ristiviitteistä (luku 4.10). Ne toimivat sisäisesti niin, että dokumentin ensimmäisellä kääntökerralla kirjoitetaan väliaikaistiedostoon muistiin tarpeellisia asioita ja seuraavalla kääntökerralla hyödynnetään väliaikaistiedostoa. Dokumentin sivumäärän ja viimeisen sivunumeron tallentamiseen on olemassa paketti `totpages`⁸, joka lisää viimeiselle sivulle automaattisesti ristiviitteen nimeltä `TotPages`. Sivumäärän ja viimeisen sivun numeron voi latio komennoilla `\ref` ja `\pageref`. Esimerkki 2.4 havainnollistaa niiden käyttöä.

Muunlaisten kokonaislaskurien toteuttamiseen voi käyttää `totcount`⁹-pakettia. Se tarjoaa komennon, jolla rekisteröidään olemassa oleva laskuri kokonaislaskuriksi. Lisäksi on komento, jolla ladotaan tai palauteaan laskurin lopullinen arvo.

⁸ <https://www.ctan.org/pkg/totpages>

⁹ <https://www.ctan.org/pkg/totcount>

```

1 \documentclass{article}
2 \usepackage{totpages}
3
4 \begin{document}
5 Sivumäärä: \ref{TotPages}. Viimeinen sivu: \pageref{TotPages}.
6
7 \addtocounter{page}{10} % Sivunumerot 11, 12, 13...
8 \newpage Jotain... \newpage ...sisältöä.
9 \end{document}

```

Esimerkki 2.4: Dokumentin sivumäärän ja viimeisen sivun numeron latominen.

```

\newcounter{oma} % Luodaan laskuri "oma".
\regtotcounter{oma} % Rekisteröidään "oma" kokonaislaskuriksi.
\addtocounter{oma}{1} % Kasvatetaan laskurin arvoa.
\total{oma} % Ladotaan laskurin lopullinen arvo.
\totvalue{oma} % Palautetaan laskurin lopullinen arvo.

```

Edellä olevan esimerkin viimeistä komentoa `\totvalue` ei käytetä laskurin arvon latomiseen itse dokumenttiin vaan sitä käytetään teknisissä yhteyksissä kuten toisen Latex-komennon argumenttina.

Myös Latexin valmiita laskureita voi rekisteröidä kokonaislaskureiksi. Tässä oppaassa on yhteensä 5 numeroitua päälukua, ja mainittu lukumäärä saatiin laskettua seuraavasti:

```

\regtotcounter{chapter} % Sijoitetaan dokumentin esittelyosaan.
\total{chapter} % Latoo päälukujen (chapter) lukumäärän.

```

2.6 Laatikot

Tex ajattelee ladottavan tekstin ja muun sisällön laatikoina eli suorakulmioina, joilla on leveys ja korkeus. Esimerkiksi kirjaimet ovat teknisesti laatikoita, joita ladotaan peräkkäin. Samoin monet valmiit ympäristöt luovat laatikon, joka sisältää jotakin ja vie tietyn verran tilaa sivulta.

Dokumentin kirjoittajalle on välillä hyötyä luoda omia laatikoita, koska laatikko toimii yhtenä kokonaisuutena, jonka reunoja muut elementit ja latomiskoneisto kunnioittavat.

2.6.1 Pienet laatikot

Yksinkertainen peruslaatikko tehdään `\mbox`-komennolla, joka latoo argumenttina annetun tekstin näkymättömän laatikon sisään. Näin luotu laatikko pysyy koossa, eli sen sisältämiä sanoja ei esimerkiksi tavuteta. Komento sopiikin hyvin tavutuksen estämiseen yksittäisessä tilanteessa.

Toinen laatikkokomento `\makebox` toimii samalla tavalla kuin `\mbox` mutta hyväksyy yhden tai kaksi valinnaista argumenttia. Komentojen muoto on seuraava:

```
\makebox[leveys]{teksti}
\makebox[leveys][sijainti]{teksti}
```

Argumentti `leveys` on laatikon leveysmitta ja `sijainti` on kirjainkoodi, joka ilmaisee laatikon sisällön vaakasuuntaisen sijoittelun. Kirjain `c` (center) keskittää sisällön, `l` (left) tasaa sisällön vasemmalle, `r` (right) tasaa sisällön oikealle ja `s` (stretch) tasaa sisällön molemmista reunoista eli venyttää laatikon sisäisiä sanavälejä sopivasti.

```
\makebox[12em][s]{Laatikossa on tilaa.}
```

⇒ Laatikossa on tilaa.

`\makebox`-komennon `leveys`-argumentissa voi käyttää apuna mittoja `\width` (`leveys`), `\height` (`korkeus`), `\depth` (`syvyys`) ja `\totalheight` (`kokonaiskorkeus: \height + \depth`), jotka määräytyvät laatikon sisällön luontaisten mittojen perusteella. Kuvassa 2.1 (s. 45) oleva typografinen viivasto havainnollistaa näitä mittoja.

Seuraava esimerkki latoo laatikon, jonka leveys on 1,5-kertainen sisällön luontaiseen leveyteen nähden, ja keskittää tekstin. Pystyviivat ovat mukana rajojen hahmottamisen vuoksi.

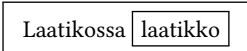
```
|\makebox[1.5\width][c]{Keskitetty}|
```

⇒ | Keskitetty |

Laatikkokomento `\fbox` toimii kuten `\mbox` mutta piirtää laatikolle myös näkyvät kehykset. Komento `\framebox` on puolestaan kehyselinen versio `\makebox`-komennosta. Kehysviivan leveyden voi asettaa

mitan `\fboxrule` avulla, ja kehysviivan etäisyys sisällöstä määritellään mitassa `\fboxsep`.


```
\setlength{\fboxsep}{3bp}
\setlength{\fboxrule}{.5bp}
\framebox[1.2\width][c]{Laatikossa \fbox{laatikko}}
```

⇒ 

Värikköisten laatikoiden ja kehysten tekemistä neuvotaan luvussa 4.2.7, joka käsittelee väriasioita muutenkin.

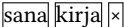
Tekstiä sisältävien laatikoiden korkeus vaihtelee kirjainten ja muiden merkkien muodon mukaan. Esimerkiksi kirjainten ala- ja yläpidennykset kasvattavat laatikon kokoa gemenalinjan yläpuolelle tai peruslinjan alapuolelle. Seuraavasta esimerkistä voi vertailla tekstilaatikoiden muodostumista:

```
\setlength{\fboxsep}{0bp}
\fbox{sana} \fbox{kirja} \fbox{*}
```

⇒ 

Edellisen esimerkin kaltaisissa tilanteissa voi olla tarpeen käyttää laatikon sisällä komentoa `\strut`. Se latoo näkymättömän, leveydettömän merkin, jonka korkeus on rivikorkeuden mukainen (mitta `\baselineskip`).

```
\setlength{\fboxsep}{0bp}
\fbox{\strut sana} \fbox{\strut kirja} \fbox{\strut *}
```

⇒ 

2.6.2 Suuret laatikot

Jos laatikon täytyy sisältää useita tekstikappaleita tai muuta pystysuuntaista sisältöä, tarvitaan esimerkiksi `\parbox`-komentoa. Se tarvitsee ainakin kaksi argumenttia: laatikon leveyden ja sisällön.

```
\parbox{leveys}{sisältö}
```

`\parbox`-laatikon sisällä teksti ladotaan samoin kuin teksti normaalitikin eli rivit katkaistaan sanavälien kohdalta ja mahdollisesti tavutuskohdista. Muutkin tekstikappaleen asetukset pätevät, ja asetukset voi määrittää haluamikseen laatikon sisällä. Lisätietoa tekstikappaleista on luvussa 4.1.

Komento `\parbox` hyväksyy myös yhden, kaksi tai kolme valinnaisista argumenttia. Niillä määritellään laatikon pystysuuntainen sijoittelu ympäristöönsä nähden, laatikon korkeus ja laatikon sisällön pystysuuntainen sijoittelu. Valinnaiset argumentit toimivat seuraavasti:

```
\parbox[sijainti]{leveys}{sisältö}
\parbox[sijainti][korkeus]{leveys}{sisältö}
\parbox[sijainti][korkeus][sisäsijainti]{leveys}{sisältö}
```

Argumentti `sijainti` on kirjain `c` (center, keskilinja), `t` (top, ylälinja) tai `b` (bottom, alinaja). Näillä määritetään, miten laatikko sijoitetaan pystysuunnassa suhteessa tekstiin. Seuraava esimerkki havainnollistaa ensimmäisen valinnaisen argumentin toimintaa. Laatikon sisällä on rivinvaihtokomentoja (`\`) ja kolme tekstiriviä.

```
abc \parbox[c]{1em}{1 \ 2 \ 3}
abc \parbox[t]{1em}{1 \ 2 \ 3}
abc \parbox[b]{1em}{1 \ 2 \ 3} abc
```

$$\begin{array}{cccc} & & 1 & \\ & & 2 & \\ \Rightarrow & abc & 2 & abc & 1 & abc & 3 & abc \\ & 3 & & 2 & & & & \\ & & & 3 & & & & \end{array}$$

Komennon toinen valinnainen argumentti `korkeus` määrittää laatikon korkeuden. Oletuksena `korkeus` määräytyy sisällön korkeuden mukaan, mutta tällä argumentilla `korkeus` asetetaan kiinteästi. Se voi olla hyödyllistä varsinkin silloin, kun sisältö halutaan tasata laatikon sisällä ylös, keskelle tai alas.

Kolmas valinnainen argumentti `sisäsijainti` nimittäin asettaa laatikon sisällön pystysuuntaisen tasauksen. Argumentiksi annetaan kirjain `c` (center, keskelle), `t` (top, ylös), `b` (bottom, alas) tai `s` (stretch, ylös ja alas). Viimeksi mainittu eli ylös ja alas tasaaminen vaatii, että laatikon sisällä on venyviä pystysuuntaisia välejä. Niitä käsitellään luvussa 2.4.3.

Ympäristö `minipage` vastaa toiminnallisesti `\parbox`-komentoa. Se hyväksyy täsmälleen samat argumentit. Erona on se, että `minipage`-ympäristön sisällä voi käyttää laatikon omia alaviitteitä (luku 4.11), jotka ladotaan laatikon alaosaan.

```
\begin{minipage}[sijainti][korkeus][sisäsijainti]{leveys}
...
\end{minipage}
```

2.6.3 Laatikoiden siirtely

Komennolla `\raisebox` voi latoa laatikon tekstin ylemmäs tai alemmas kuin se normaalisti sijoittuisi. Komennon ensimmäinen argumentti on etäisyysmitta, kuinka paljon ylemmäs teksti ladotaan. Negatiivinen mitta vie tekstin alemmas. Toinen argumentti on laatikon teksti.

```
\raisebox{etäisyysmitta}{teksti}
```

Argumentissa etäisyysmitta voi käyttää apuna mittoja `\width`, `\height`, `\depth` ja `\totalheight`, joiden merkitys on sama kuin `\makebox`-komennolla (luku 2.6.1). `\raisebox`-komennolle voi antaa yhden tai kaksi valinnaista argumenttia, joilla määritellään laatikolle uusi korkeus ja syvyys. Molemmat ovat mittoja, ja ne korvaavat siirretyn laatikon tekstin luonnollisen korkeuden ja syvyyden. Komennon argumentit annetaan seuraavasti:

```
\raisebox{etäisyysmitta}[korkeus]{teksti}
\raisebox{etäisyysmitta}[korkeus][syvyys]{teksti}
```

Grafiikkaan kuten kuviin ja väreihin erikoistunut paketti `graphicx`¹⁰ sisältää hyödyllisiä laatikkokomentoja: `\scalebox`, `\resizebox` ja `\rotatebox`. Komento `\scalebox` skaalaa tekstin eri kokoiseksi vaak- tai pystysuunnassa.

```
\scalebox{vaakaskaalaus}[pystyskaalaus]{teksti}
```

Argumentit vaakaskaalaus ja pystyskaalaus ovat kertoimia. Pystysuuntainen skaalausargumentti on valinnainen: jos sen jättää pois, pystysuuntainen skaalaus on samansuuruinen kuin vaakasuuntainen. Negatiivisella kertoimella voi kääntää tekstin toisinpäin.

¹⁰ <https://www.ctan.org/pkg/graphicx>


```
\scalebox{2}[1]{leveä} \scalebox{-1}[1]{käännetty}
```

⇒ **leveä** *ytännössä*

Samantapainen skaalaustoiminto saadaan myös komennolla `\resizebox`. Sen argumentiksi ei anneta kertoimia vaan mittoja, eli teksti skaalataan annettuihin mittoihin.

```
\resizebox {vaakamitta}{pystymitta}{teksti}  
\resizebox*{vaakamitta}{pystymitta}{teksti}
```

Komennon normaali versio käsittelee pystymitta-argumentin siten, että se tulee olemaan tekstin korkeus eli kirjainten peruslinjan yläpuolinen osa. Komennon tähdellinen versio puolestaan toimii siten, että pystymitta on tekstin kokonaiskorkeus eli peruslinjan yläpuolinen osa (korkeus) ja alapuolinen osa (syvyys) yhdessä. Kuva 2.1 (s. 45) havainnollistaa asiaa. `\resizebox`-komennon argumenteissa voi käyttää apuna mittoja `\width`, `\height`, `\depth` ja `\totalheight` samoin kuin `\makebox`-komennossa (luku 2.6.1).

Laatikoiden pyörittely käy `\rotatebox`-komennolla. Sen argumentiksi täytyy antaa ainakin kääntökulma asteina ja käännettävä teksti.

```
\rotatebox[asetuksia]{kääntökulma}{teksti}
```

Valinnainen argumentti `asetuksia` voi sisältää asetusvalitsimia ja niiden arvoja. Tärkein valitsin on `origin`, jolla määritetään käännettävän tekstin kääntöpiste. Valitsimen arvoksi annetaan yksi tai kaksi seuraavista kirjaimista: `c` (center, keski), `l` (left, vasen), `r` (right, oikea), `t` (top, ylä), `b` (bottom, ala) tai `B` (baseline, peruslinja).

```
\rotatebox[origin=l] {25}{ylämäki}  
\rotatebox[origin=r]{-25}{alamäki}
```

⇒ *ylämäki* *alamäki*

`\rotatebox`-komennon valinnaisen argumentin valitsin `units` määrittelee käytetyn kulmayksikön. Valitsimelle annetaan arvoksi täysympyrän yksikkömäärä, ja oletusasetus on `units=360`. Jos haluaa käyttää radiaaneja, sopiva asetus on `units=6.283185` (2π rad).

Luku 3

Asetukset

Dokumentin yleiset asetukset koostuvat Latexissa dokumenttiluokan valinnasta, sivun koon ja marginaalien määrittelystä, fonttien määrittelystä sekä kieliasetuksista. Näitä kaikkia käsitellään tässä luvussa.

Kirjoittajan ei tarvitse tehdä kaikkia dokumentin asetuksia kerralla. On ehkä jopa suotavaakin keskittyä aluksi lähinnä dokumentin sisällön ja rakenteen suunnitteluun ja tuottamiseen. Ulkoasuun kyllä ehtii tulla kirjoittamisen aikana monenlaisia ajatuksia, joita ei olisi välttämättä alussa osannut huomioida. Kirjoittamaan pääsee hyvinkin yksinkertaisen dokumenttirungon avulla (esimerkki 1.2, s. 19), mutta tässä luvussa käsitellään asetusten määrittelyä ja mahdollisuuksia melko perusteellisesti.

3.1 Dokumenttiluokat

Latexin lähdedokumenttien alussa on aina samankaltainen rivi, joka määrittelee käytettävän dokumenttiluokan ja mahdollisesti dokumentin perusasetuksia. Dokumenttiluokka määritellään komennolla `\documentclass` ja sen argumentiksi annetaan dokumenttiluokan nimi. Valinnaisilla argumenteilla vaikutetaan asetuksiin.

```
\documentclass[a4paper, 12pt]{article}
```

Dokumenttiluokka on eräänlainen pohjadokumentti eli ominaisuuksien ja asetusten kokoelma, jonka varaan oma dokumentti kirjoitetaan.

Eri dokumenttiluokat sisältävät erilaisia ominaisuuksia ja erilaiset oletusasetukset. Edellä olevassa esimerkissä käytettiin `article`-luokkaa, joka on yleiskäyttöinen luokka monenlaisten dokumenttien kirjoittamiseen. Seuraavissa alaluvuissa käsitellään tavallisimpia dokumenttiluokkia ja niiden asetuksia.

3.1.1 Perusdokumenttiluokat

Muutama dokumenttiluokka kuuluu Latexin perusvalikoimaan, eli ne ovat aina saatavilla ja asennettuna. Niitä kutsutaan välillä standardeiksi dokumenttiluokiksi, ja ne ovat ikään kuin muuttumattomia perusluokkia, joiden varaan on turvallista rakentaa omia dokumentteja tai muita dokumenttiluokkia. Muita dokumenttiluokkia kehitetään Latexin perusosien ulkopuolella, ja niiden ominaisuudet voivat muuttua ja kehittyä nopeammin ajan myötä.

Normaalit paperisivuihin perustuvat dokumentit tehdään luokkien `article`, `report` tai `book` avulla. Ne ovat keskenään hyvin samanlaisia, mutta niiden oletusasetukset poikkeavat toisistaan. Karkeasti jaoteltuna `article`-luokka on tarkoitettu lyhyehköille artikkeleille ja yleiseksi perustaksi monenlaisille dokumenteille. Sen sijaan `report` ja `book` on tarkoitettu laajoihin dokumentteihin, ja niissä on kirjatyypografian piirteitä.

Suurin edellä mainittujen perusluokkien ero on otsikoinnissa ja dokumentin jäsentämisessä: `report` ja `book` sisältävät suuret pääluvut (`\chapter`) eli otsikot, jotka alkavat uudelta sivulta; `article` ei sisällä samanlaisia päälukuja vaan ainoastaan tavalliset leipätekstin mukana kulkevat väliotsikot (`\section`, `\subsection` jne.). Tekstin jäsentämistä käsitellään luvussa 4.4 ja otsikointia tarkemmin luvussa 4.4.2.

Toinen ero perusluokkien välillä on se, että vain `book`-luokka sisältää komennot `\frontmatter`, `\mainmatter` ja `\backmatter`, joita voi käyttää ilmaisemaan tietokirjan erityyppiset osat: esittely-, sisältö- ja liitesivut. Näitä käsitellään luvussa 4.4.5. Lisäksi `report`-luokka sisältää ympäristön `abstract`, joka on tarkoitettu tutkimusraportin tai vastaavan dokumentin tiivistelmäosan tekemiseen; `book`-luokassa sitä ei ole.

Muut erot koskevat lähinnä oletusasetuksia, mutta asetukset ovat tietenkin muutettavissa, eivätkä ne siten ole määräävä tekijä valittaessa

a4paper	landscape	openright	10pt
a5paper	final	openany	11pt
b5paper	draft	onecolumn	12pt
executivepaper	oneside	twocolumn	titlepage
legalpaper	twoside	fleqn	notitlepage
letterpaper	openbib	leqno	clock

Taulukko 3.1: Perusdokumenttiluokkien valitsimet.

luokkien `article`, `report` ja `book` välillä. Dokumenttiluokkien asetuksia käsitellään luvussa 3.1.2.

Muita perusdokumenttiluokkia ovat `letter` ja `slides`. Nimensä mukaisesti `letter` on kirjeiden kirjoittamista varten. Se sisältää kirjeen tyyppilliseen rakenteeseen liittyviä komentoja ja soveltuu varsinkin tarpeisiin, joissa täytyy automaattisesti tuottaa samanmuotoisia kirjeitä usealle vastaanottajalle. Sama Latex-dokumentti voi sisältää useita kirjeitä, ja sivunumerointi ja alaviitteiden numerointi alkaa joka kirjeessä alusta – kuten on tietysti toivottavaa. `letter`-luokkaa käsitellään tarkemmin luvussa 5.2.

Piirtoheittimien läpinäkyvien kalvojen ja sen kaltaisten dokumenttien tekemiseen on dokumenttiluokka `slides`. Piirtoheittimet ovat jääneet menneisyyteen, ja nykyään jokseenkin vastaavanlainen rooli on esitysgraafikkaohjelmilla, joilla tehdään diaesityksiä esimerkiksi esitelmien havaintomateriaaliksi. `slides`-luokkaa ei varmaan tarvita enää mihinkään, mutta esitysgraafiikkaan on olemassa erittäin monipuolinen luokka `beamer`, jonka perusasioita käsitellään luvussa 5.1.

3.1.2 Perusdokumenttiluokkien asetukset

Taulukkoon 3.1 on koottu perusdokumenttiluokkien asetusten valitsimet. Sivun koko asetetaan valitsimilla `a4paper` (210 × 297 mm), `a5paper` (148 × 210 mm), `b5paper` (176 × 250 mm), `executivepaper` (7,25 × 10,5 tuumaa), `legalpaper` (8,5 × 14 tuumaa) tai `letterpaper` (8,5 × 11 tuumaa). Latexin oletuksena on `letterpaper`, mutta oletusta voi olla muutettu Latexin jakelupaketissa. Varminta on aina itse kirjoittaa haluttu sivukoko dokumentin asetuksiin.

Sivu on oletuksena pystyasennossa, mutta valitsin landscape asettaa sen vaaka-asentoon. Tämä asetus koskee dokumentin kaikkia sivuja, mutta jos haluaa asettaa vain yksittäisiä sivuja vaaka-asentoon, täytyy käyttää `pdfscape`¹-pakettia ja sen tarjoamaa ympäristöä.

Sivun koon ja muitakin mittoja voi määrittellä monipuolisemmin `geometry`-paketin toimintojen avulla (luku 3.2). Jos sitä pakettia käyttää, ei näitä dokumenttiluokan sivukokoasetuksia tarvita välttämättä lainkaan.

Leipätekstin fontin koon voi määrittää valitsimilla 10pt, 11pt ja 12pt, joista ensin mainittu on oletus. Lyhenne pt tarkoittaa typografista pistemittaa, joita käsitellään luvussa 2.4. Nämä asetukset vaikuttavat myös sivun marginaaleihin, koska fontin koko vaikuttaa tekstiriville mahtuvien merkkien määrään ja Latex pyrkii pitämään rivit sopivan mittaisina lukemisen kannalta.

Vain kolme eri fonttikokoa leipätekstille on kovin vähän ja peräisin ajalta, jolloin Latexin fontit olivat pikseligrafiikkaa eli muodostuivat erillisistä kuvapisteistä. Nykyaikana fontit ovat vektorigrafiikkaa eli matemaattisten kaavojen avulla muodostettavia kuvia, ja ne voi veyntää mihin kokoon tahansa. Sen vuoksi nämä dokumenttiluokkien fonttikokovalitsimet eivät ole kovin tarpeellisia eivätkä riitä kaikkiin tarpeisiin nykyaikana. Voi niitä silti käyttää, jos niiden avulla saa halutun lopputuloksen. Fontteja käsitellään tarkemmin luvussa 3.4.

Dokumentista voi ladata luonnosversion käyttämällä dokumenttiluokan valitsinta `draft`. Luonnoksena ladottuun dokumenttiin tulevat merkityksi (musta suorakulmio) ainakin tekstipalstan yli pursuavat rivit, mikä voi auttaa tekstikappaleiden ulkoasun korjailussa (luku 4.1). Eri paketit voivat hyödyntää `draft`-valitsinta omalla tavallaan eli muuttaa toimintaansa sen perusteella. Esimerkiksi grafiikkaan liittyvä paketti `graphicx` (luku 4.15) jättää luonnosversiosta kuvat pois, ja niiden paikalla on vain suorakulmio. Pdf-tiedoston ristiviitteistä huolehtiva `hyperref`-paketti jättää pdf:n sisäiset ristiviitteet ja sisällysluettelon tekemättä (luku 4.10). Latominen voi nopeutua huomattavasti.

¹ <https://www.ctan.org/pkg/pdfscape>

Lopullinen ladontatila on `final`, joka tarkoittaa, että dokumenttiin ladotaan kaikki sisältö ja ominaisuudet niin kuin se on tarkoitettu julkaistavaksi. Tätä valitsinta ei tarvitse käyttää, koska se on oletustila.

Sivujen yksipuolisuus (`oneside`) ja kaksipuolisuus (`twoside`) ovat asetuksia, jotka vaikuttavat sivun marginaalien asetuksiin ja mahdollisesti uuden pääluvun (`\chapter`) aloitukseen. Oletus on yksipuolinen dokumentti kaikissa muissa perusdokumenttiluokissa paitsi `book`-luokassa, jonka oletus on kaksipuolinen.

Yksipuolisessa dokumentissa kaikkien sivujen marginaalit ovat lähökohtaisesti samanlaisia ja asetuksissa puhutaan esimerkiksi vasemmasta (`left`) ja oikeasta (`right`) marginaalista. Kaksipuolisessa dokumentissa sivut ajatellaan pareittain: kaksi sivua muodostaa kirjan aukeaman, jonka vasemmalla puolella on parillinen sivunumero ja oikealla pariton. Marginaalien asetuksissa puhutaan sisämarginaaleista (`inner`) ja ulkomarginaaleista (`outer`). Sisämarginaalit ovat oletuksena pienemmät, koska niitä on aukeamalla aina kaksi vierekkäin. Ulkomarginaaleissa on tilaa marginaalihuomautuksille (luku 4.1.8). Marginaaleja ja muita sivun asetuksia käsitellään tarkemmin luvussa 3.2.

Jos dokumentti on määritetty kaksipuoliseksi (`twoside`), voidaan `openright`-valitsimella määrittää pääluvut (`\chapter`) alkamaan aina oikeanpuoleiselta sivulta. Se onkin oletus dokumenttiluokassa `book`. Valitsin `openany` asettaa pääluvut alkamaan miltä tahansa seuraavalta sivulta.

Valitsin `openbib` liittyy automaattiseen lähdeluettelon muotoiluun. Jos tämä valitsin on mukana, automaattisessa lähdeluettelon muotoilussa lisätään rivinvaihtoja lähdemerkinnän eri osien kuten teoksen tekijöiden ja teoksen nimen jälkeen. Lähdemerkintöjä ja lähdeluettelon laatimista käsitellään luvussa 4.13.

Matemaattisten kaavojen ladontaan liittyvässä `equation`-ympäristössä kaavat ladotaan normaalisti vaakasuunnassa sivun keskelle. Dokumenttiluokan valitsinta `f1eqn` käyttämällä kaavat ladotaan kuitenkin sivun vasempaan reunaan. Kaavojen numerot sijoitetaan oletuksena sivun oikeaan reunaan, mutta valitsimella `1eqno` ne saa ladottua sivun vasempaan reunaan. Matematiikkatilaa käsitellään luvussa 4.16.

Nimiösivun tai dokumentin perustietojen esitystapaan vaikutetaan valitsimilla `titlepage` ja `notitlepage`. Latexissa on yksinkertaiset komennot dokumentin perustietojen eli pääotsikon, tekijöiden nimien ja päiväyksen latomiseen (luku 4.4.1), ja näillä valitsimilla vaikutetaan siihen, ladotaanko perustiedot omalle sivulleen (`titlepage`) vaiko ensimmäisen sisältösivun alkuun (`notitlepage`). Oletusasetus vaihtelee eri dokumenttiluokissa.

Oletuksena sivun tekstit ladotaan yhdelle palstalle (`onecolumn`), mutta valitsimella `twocolumn` teksti ladotaan kahdelle palstalle. Nämä valitsimet vaikuttavat myös sivun marginaaleihin. Latexin perusosat eivät hallitse useampaa kuin kaksi palstaa, mutta `multicol`-paketin avulla saa enemmänkin. Palstoja käsitellään luvussa 4.12.

Dokumenttiluokassa `slides` toimii `clock`-valitsin, joka latoo kellonajan piirtoheitinkalvon loppuun `\note`-komennon yhteydessä. Tämä valitsin kuuluneeseen samaan paikkaan kuin piirtoheitimet ja `slides`-luokakin: kierrätykseen tai kaatopaikalle.

3.1.3 Muita dokumenttiluokkia

Kuka tahansa voi tehdä ja julkaista omia dokumenttiluokkia, ja niinpä perusdokumenttiluokkien lisäksi onkin saatavilla monenlaisia luokkia. Tässä käsitellään vain joitakin.

Luokka `memoir` on Latexin `book`-luokkaan perustuva, ominaisuuksiltaan laajennettu luokka, joka sopii yleiskäyttöön. Tähän luokkaan on sisällytetty monia dokumentin rakenteeseen ja ulkoasuun liittyviä toimintoja, joihin yleensä tarvitaan erillisiä paketteja tai muuten lisäkoodia. Sen sijaan, että ratkoi dokumentin rakenteen ja asettelun kysymyksiä erillisten pakettien avulla, saattaisi olla helpompaa lukea (lähes) pelkästään vain `memoir`-luokan ohjekirjaa ja toteuttaa asiat yhtenäisen järjestelmän avulla.

`beamer`-luokka on monipuolinen ja huippuunsa viritetty luokka esitysgrafiikan eli diaesitysten tekemiseen. Se on eräänlainen Latexin vastine esimerkiksi Libreoffice Impress- ja Microsoft Powerpoint -ohjelmille.

Pienempiin, rajattuihin erityistarpeisiin on olemassa useita luokkia. Esimerkiksi yksisivuisten taitettavien lehtisten kuten tapahtuman

käsiohjelmien tekemiseen soveltuu `leaflet`-luokka. CD-levyjen kansia voi tehdä `cd`-luokan avulla. Kuvitettuja seinäkalentereita vartenkin on tehty oma luokkansa, `wallcalendar`. Kitaran tablatuureja eli otelautakuvaan perustuvia nuotteja voi tehdä `guitartabs`-luokan avulla.

Kaikenlaista on siis olemassa. Saatavilla oleviin dokumenttiluokkiin voi tutustua esimerkiksi Comprehensive Tex Archive Network (CTAN) -verkkosivun avulla.²

3.2 Sivu

Latex-dokumentit muodostuvat aina peräkkäisistä sivuista, eli taustalla kummittelee paperiarkkeihin perustuva julkaisumuoto, vaikka dokumenttia ei varsinaisesti tulostettaisi tai painettaisi paperille. Tämän vuoksi melkein aina täytyy määrittää sivun asetukset kuten paperikoko ja marginaalit. Ne tehdään `geometry`-paketin avulla, jota käsitellään luvussa 3.2.1. Joskus halutaan muokata sivun ylä- tai alatunnistetta eli verrattain pysyviä tunnistetietoja, jotka toistuvat joka sivulla ylä- tai alareunassa. Niiden muokkaamiseen käytetään `fancyhdr`-pakettia, jonka ohjeita on luvussa 3.2.4.

3.2.1 Sivun koko ja marginaalit

Paperin eli sivun kokoon ja marginaaleihin pääsee vaikuttamaan `geometry`³-paketin avulla. Halutut asetukset voi kertoa paketin lataamisen yhteydessä. Seuraavassa esimerkissä asetetaan paperikoko (a4paper eli A4-koko), ylä- ja alamarginaalin pituus (`top`, `bottom`) sekä vasemman ja oikean marginaalin pituus (`left`, `right`).

```
\usepackage[a4paper, top=20mm, bottom=30mm,
left=20mm, right=20mm]{geometry}
```

Vaihtoehtoisesti asetukset voi ilmaista erikseen `\geometry`-komennon avulla seuraavalla tavalla:

```
\usepackage{geometry}
\geometry{a4paper, top=20mm, bottom=30mm, left=20mm, right=20mm}
```

² <https://www.ctan.org/topic/class>

³ <https://www.ctan.org/pkg/geometry>

Mikäli myöhemmin dokumentissa täytyy vaihtaa sivun asettelua, käytetään komentoa `\newgeometry`. Alkuperäiset asetukset palautetaan komennolla `\restoregeometry`.

```
\newgeometry{top=..., bottom=..., ...} % Uudet asetukset.  
\restoregeometry % Palautetaan alkuperäiset asetukset.
```

Valmiiksi määriteltyjä standardipaperikokoja on useita. ISO-standardin mukaiset koot A0–A6 valitaan valitsimilla kuten `a3paper`, `a4paper` tai `a5paper`. Samoin käytetään myös kokoja B0–B6 ja C0–C6, esimerkiksi `b2paper` tai `c6paper`. Lisäksi on valittavissa useita japanilaisia ja yhdysvaltalaisia standardeja käyttämällä valitsimia kuten `b0j–b6j`, `ansi-a-paper` tai `letterpaper`.

Mikäli standardit koot eivät riitä, voi sivun mitat määrittää vapaasti `papersize`-valitsimella, jolle annetaan arvoksi leveys- ja korkeusmitta.

```
\geometry{papersize={10cm, 16cm}}
```

Sivu on oletuksena pystyasennossa (`portrait`), mutta sen voi asettaa vaakasentoon valitsimella `landscape`. Tämä asetusta on voimassa koko dokumentin ajan. Jos sen sijaan haluaa asettaa vain yksittäisiä sivuja vaakasentoon, täytyy käyttää `pdflscape`⁴-pakettia ja sen tarjoamaa `landscape`-ympäristöä. Ympäristön sisältö ladotaan vaakasuuntaisille sivuille.

```
\begin{landscape}  
...  
\end{landscape}
```

Sivun asetusten suunnittelussa voi olla avuksi valitsin `showframe`, joka piirtää viivat marginaalien kohdalle ja merkitsee myös ylä- ja alatunnisteiden sekä marginaalihuomautusten rajoja.

Taulukkoon 3.2 on koottu tärkeimpiä valitsimia marginaalien mittojen asettamiseksi. Valitsimille annetaan arvoksi Texin mittayksikkö, jotka ovat taulukossa 2.7 (s. 43). Kaksipuolisessa asettelussa (`twoside=true`) sivut muodostavat aukeaman eli on erikseen vasemmanpuoleinen

⁴ <https://www.ctan.org/pkg/pdflscape>

Valitsin	Merkitys	
left	inner	vasen marginaali tai sisämarginaali
right	outer	oikea marginaali tai ulkomarginaali
top		ylämarginaali
bottom		alamarginaali

Taulukko 3.2: *geometry*-paketin valitsimia sivun marginaalien määrittelemiseen.

Valitsin	Merkitys
hscale	tekstialueen osuus sivun leveydestä
vscale	tekstialueen osuus sivun korkeudesta
hmarginratio	vasemman (sisä) ja oikean (ulko) marginaalin suhde
vmarginratio	ylä- ja alamarginaalin suhde

Taulukko 3.3: *geometry*-paketin valitsimia sivun teksti-tilan ja marginaalien suhteiden määrittämiseen. Osuudet (*scale) ilmaistaan prosenttikertoimella (esim. 0.7). Suhteet (*marginratio) ilmaistaan suhdelukuna (esim. 2:3).

ja oikeanpuoleinen sivu. Tällöin vasen ja oikea marginaali vuorottelevat, ja niitä on havainnollisempaa kutsua sisä- ja ulkomarginaaliksi. Marginaalien määrittelyssä voi käyttää valitsimen left sijasta valitsinta inner, ja vastaavasti right-valitsin voidaan korvata outer-valitsimella. Teknisesti näillä ei ole mitään eroa.

Sivun tekstialueen koon ja marginaalit voi määrittää myös suhteellisesti. Voisi esimerkiksi määrittää, että leveyssuunnassa tekstialue täyttää 0,7-kertaisesti (70 %) sivun leveyden ja loput jää marginaaleille. Marginaalien keskinäiset suhteetkin voi ilmaista suhdelukuna: esimerkiksi vasemman (sisä) ja oikean (ulko) marginaalin suhde voisi olla 2:3. Suhteellisessa tavassa ei tarvitse ottaa kantaa sivun kokoon eikä muihinkaan varsinaisiin mittoihin, vaan samat tekstialueen ja marginaalien suhteet säilyvät, vaikka sivukokoa muuttaisikin. Suhteellisia mittoja koskevia valitsimia on koottu taulukkoon 3.3.

Klassisessa kirjatypografiassa, jossa teksti ladotaan yhdelle palstalle, marginaalien suuruusjärjestys on suurimmasta pienimpään seuraavanlainen: ala-, ulko-, ylä- ja sisämarginaali. Marginaalien suhdeluvut ovat samassa järjestyksessä: 32, 28, 20, 17. *geometry*-paketin valitsimilla tämä ilmaistaan seuraavasti:

`\geometry{hmarginratio=17:28, vmarginratio=20:32}`

Klassisia suhteita ei nykyaikana yleensä noudateta kovin tarkasti, mutta niistä kannattaa ymmärtää yleinen ajatus. Alamarginaalin pitäisi olla hieman suurempi kuin ylämarginaali, koska muuten tekstialue tuntuu pudonneen sivulla alas. Alamarginaalissa (alatumnisteessa) on yleensä sivunumero. Kirjan sivuja katsotaan pareittain eli aukeamina, joten sisämarginaaleja on kaksi vierekkäin. Siksi sisämarginaalit yksittäin ajateltuna ovat pienemmät kuin ulkomarginaalit. Lisäksi ulkomarginaaleja voidaan käyttää huomautusten kirjoittamiseen, mikä on melko yleinen käytäntö tietokirjoissa.

Marginaalihuomautukset (luku 4.1.8) sijaitsevat oletuksena sivujen ulkomarginaalissa tai oikeanpuoleisessa marginaalissa. Huomautuspalstan leveys asetetaan valitsimella `marginparwidth`, ja palstan etäisyys sivun varsinaisesta tekstialueesta määritellään valitsimella `marginparsep`. Kumpikin valitsin tarvitsee argumentiksi mitan. Jos haluaa vaihtaa huomautukset sivun vastakkaiseen marginaaliin, lisätään mukaan valitsin `reversemarginpar`.

Perus LaTeX osaa latoa tekstin yhdelle tai kahdelle palstalle, ja `geometry`-paketin valitsimella `onecolumn` tai `twocolumn` asetetaan, kumpi tila on oletuksena päällä. Useampikin palsta on mahdollista laajennuspaketin avulla. Valitsimen `columnsep` avulla asetetaan palstojen välinen etäisyys. Käytännössä tämä valitsin on asettaa mitan `\columnsep`, jota voi muokata myös komennolla `\setlength`, kuten muitakin mittoja (luku 2.4). Tarkempaa tietoa palstoista on luvussa 4.12.

3.2.2 Sivun mittoja

Sivun kokoasetusten määrittämisen jälkeenkin voi dokumentissa olla tarpeen hyödyntää joitakin sivun mittoja. Usein esimerkiksi halutaan piirtää taulukko tai kuva, joka on sivun tekstialueen levyinen tai siihen suhteutettu. Silloin on kätevää käyttää mitta, jonka arvona on juuri tekstialueen leveys. Tärkeimmät sivun mitat on koottu taulukkoon 3.4, mutta perusteellisemmin niitä käsitellään `geometry`⁵-paketin ohjekirjassa.

⁵ <https://www.ctan.org/pkg/geometry>

Mitta	Merkitys
<code>\paperwidth</code>	paperin eli sivun leveys
<code>\paperheight</code>	paperin eli sivun korkeus
<code>\textwidth</code>	tekstialueen leveys
<code>\columnwidth</code>	nykyisen palstan leveys
<code>\linewidth</code>	nykyisen rivin leveys
<code>\textheight</code>	tekstialueen korkeus

Taulukko 3.4: Sivun mittoja.

Taulukossa 3.4 mainittu mitta `\linewidth` eroaa tekstialueen (`\textwidth`) tai palstan (`\columnwidth`) leveysmitasta esimerkiksi silloin, kun tekstikappaletta on sisennetty. Kappaleiden ensimmäinen rivi voi olla sisennetty, ja sen vuoksi rivi ei ole täysilevyinen. Muutkin sisennykset kuten lohkolainaukset tai luettelmat vaikuttavat rivin leveysmittaan. Sisennysasiat liittyvät tekstikappaleiden muotoiluun, jota käsitellään luvussa 4.1. Luettelmaa puolestaan käsitellään luvussa 4.6.

3.2.3 Leikkuuvarat

Sivu voi olla ulkoisesti erikokoinen kuin sisäisesti. On siis mahdollista asettaa sivu esimerkiksi A4-kokoiseksi ja käsitellä marginaalit ja muut sivun mitat A4-koon mukaan, mutta ulkoisesti tai fyysisesti sivu onkin osana suurempaa sivua tai paperiarkkia. Tällaista tarvitaan ainakin silloin, kun halutaan merkitä leikkuuvarat dokumentin painamista varten.

Painokoneen paperiarkkien leikkauskohta ei välttämättä osu täsmälleen samaan kohtaan pdf-tiedoston sivun reunan kanssa, ja siksi dokumentissa reunaan saakka yltävät kuvat asetetaan varmuuden vuoksi hieman ylikokoiseksi. Sivun reunaan saakka aiottu kuva siis yltävä lähde-pdf:ssä pari millimetriä varsinaisen sivualueen ulkopuolelle eli leikkuuvaran puolelle. Tällä varmistetaan, että painamisen jälkeen leikatuissa paperiarkeissa kuva varmasti yltää reunaan saakka.

Jos dokumenttiin tarvitaan sivun ulkopuoliset leikkuuvarat, määritellään dokumentin ulommaiset mitat edelleen samalla tavalla kuin tavallisestikin eli esimerkiksi valitsimella `papersize` (luku 3.2.1). Sen si-

```

1 \geometry{
2   papersize={220mm, 307mm},
3   layout=a4paper,           % tai: layoutsizе={210mm, 297mm}
4   layoutoffset={5mm, 5mm},
5   showcrop
6 }

```

Esimerkki 3.1: Sivun ulkoisten ja sisäisten mittojen sekä leikkuuvaarojen määrittäminen.

jaan sivun sisäiset mitat täytyy määritellä toisella tavalla, käyttämällä valitsinta `layout` tai `layoutsizе`.

Esimerkissä 3.1 käytetään sisäisesti A4-kokoista (210 × 297 mm) sivua, mutta sivulle on määritetty joka puolelle 5 mm:n leikkuuvarat. Niinpä ulkoisesti sivu on 10 mm leveämpi ja korkeampi, eli ulkoiset mitat ovat 220 × 307 mm. Valitsimella `layoutoffset` asetetaan sisäisen sivun etäisyys ulkoisen sivun vasemmasta ylänurkasta. Esimerkissä on mukana myös valitsin `showcrop`, joka merkitsee sisäisen ja ulkoisen sivun rajakohdan eli leikkuuvaran rajan. Merkinnät näkyvät vain sivun nurkissa leikkuuvaran puolella, joten ne eivät päädy lopulliseen painotuotteeseen.

3.2.4 Ylä- ja alatunnisteet

`geometry`-paketin asetuksiin kuuluu pari valitsinta, joilla vaikutetaan ylä- ja alatunnisteiden mittoihin. Valitsimella `head` ilmaistaan ylätunnisteen korkeus ja valitsimella `headsep` sen etäisyys sivun tekstipalstasta. Alatunnisteen peruslinjan etäisyys tekstipalstasta säädetään valitsimella `footskip`. Taulukkoon 3.5 on koottu näiden valitsimien merkitys, ja seuraavassa on niiden käyttämisestä esimerkki. Mukana on myös valitsin `showframe`, joka piirtää sivulle apuviivoja. Se auttaa sivun mittojen suunnittelussa.

```
\geometry{head=24bp, headsep=8bp, footskip=12mm, showframe}
```

Latexin perusosat eivät sisällä kovin kummoista keinovalikoimaa ylä- ja alatunnisteiden muokkaamiseen, mutta pari hyödyllistä sivutyyliä on kuitenkin mukana. Ylä- ja alatunnisteet määräytyvät sivutyylin perusteella, ja haluttu tyyli asetetaan voimaan komennolla `\pagestyle`:

Valitsin	Merkitys
head	ylätunnisteen korkeusmitta
headsep	ylätunnisteen etäisyys tekstipalstasta
footskip	alätunnisteen peruslinjan etäisyys tekstipalstasta

Taulukko 3.5: `geometry`-paketin valitsimet ylä- ja alätunnisteiden mittojen asettamiseen.

```
\pagestyle{plain}
```

Edellä mainittu sivutyyli `plain` latao alätunnisteeseen sivunumeron. Se on yleensä oletustyyli. Sivunumero on peräisin laskurista `page` ja sen arvon tulostavasta komennosta `\thepage` (luku 2.5). Toinen hyödyllinen tyyli on `empty`, joka tarkoittaa tyhjää, eli ylä- eikä alätunnisteeseen ei ladota mitään.

Yksittäiselle sivulle voi asettaa muusta dokumentista poikkeavan sivutyylin komennolla `\thispagestyle`. Komento siis vaikuttaa vain sillä hetkellä ladottavan sivun tyyliin, ja sen jälkeen palataan taas voimassa olevaan tyyliin, joka on aiemmin määritelty komennolla `\pagestyle`.

```
\thispagestyle{empty}
```

Sivutyylit `headings` latao ylä- ja alätunnisteeseen aukeaman vasemmanpuoleisille sivuille esimerkiksi kirjan pääluvun nimen ja oikeanpuoleisille sivuille meneillään olevan alaluvun nimen. Mainitut lukujen nimet tulevat sivun sisäreunaan; ulkoreunaan ladotaan sivunumero.

Teknisesti ja sisäisesti tämä on toteutettu siten, että pääluvun aloitettava otsikkokomento (esim. `\chapter`) automaattisesti määrittelee joka kerta uudelleen komennon `\leftmark`, niin että se sisältää pääluvun nimen. Sivuja ladottaessa Latex sitten latao ylä- ja alätunnisteeseen sen, mitä `\leftmark`-komento sattuu tulostamaan. Vastaavasti alaluvun otsikkokomento (esim. `\section`) määrittelee uudelleen komennon `\rightmark`, niin että se sisältää alaluvun nimen. Tämän komennon tulostama teksti ladotaan aukeaman oikeanpuoleisille sivuille.

Jos haluaa itse vaikuttaa ylä- ja alätunnisteen tekstiin, voi käyttää sivutyylillä `myheadings` sekä komentoa `\markboth`, jolla määritellään aukeaman vasemmanpuoleisen ja oikeanpuoleisen sivun ylä- ja alätunnisteen teksti. Tämä

Tyyli	Merkitys
arabic	arabialaiset luvut: 1, 2, 3...
roman	roomalaiset luvut: i, ii, iii...
Roman	roomalaiset luvut: I, II, III...
alph	kirjaimet: a, b, c... (vain 1–26)
Alph	kirjaimet: A, B, C... (vain 1–26)

Taulukko 3.6: Sivunumerointityylit `\pagenumbering`-komennon argumentiksi.

komento on tarkoitettu suoritettavaksi päälukujen yhteydessä. Komennolla `\markright` määritellään pelkästään oikeanpuoleisen sivun teksti, ja komento on tarkoitettu suoritettavaksi aina alalukujen yhteydessä.

```
\markboth{vasen}{oikea} % määrittelee: \leftmark ja \rightmark
\markright{oikea}      % määrittelee: \rightmark
```

Sivunumeroinnin tyyliin voi vaikuttaa esimerkiksi komennolla `\pagenumbering`, jonka argumentiksi annetaan numerointityylin nimi. Ne on koottu taulukkoon 3.6.

```
\pagenumbering{roman}
```

Edellä mainittu komento määrittelee käytännössä uudelleen komennon `\thepage`, joka on tarkoitettu juuri sivunumerolaskurin latomiseen. Lisätietoa sivunumeroista ja muista laskureista on luvussa 2.5.

Monipuolisemmin ylä- ja alatunnisteita voi muokata paketin `fancyhdr`⁶ toimintojen avulla. Silloin sivutyylinä voi olla myös fancy:

```
\usepackage{fancyhdr}
\pagestyle{fancy}
```

Kun käytössä on sivutyyli fancy, voi ylä- ja alatunnisteiden sisällön asettaa vapaasti komennolla `\fancyhf`. Komennon argumenttien merkitys on seuraavanlainen:

```
\fancyhf[paikka]{sisältö}
```

⁶ <https://www.ctan.org/pkg/fancyhdr>

Valitsin	Merkitys
H	ylätunniste (header)
F	alätunniste (footer)
E	parillinen sivu, vasen (even)
O	pariton sivu, oikea (odd)
L	sivun vasen reuna (left)
C	sivun keskelle (center)
R	sivun oikea reuna (right)

Taulukko 3.7: `\fancyhf`-komennon valitsimia ylä- tai alätunnisteen paikan määrittämiseen.

Valinnainen argumentti paikka kertoo, mihin paikkaan tai paikkoihin sisältö sijoitetaan. Vaihtoehtoina on ylä- tai alätunniste, pariton tai parillinen sivu, sivun vasen reuna, keskiosa tai oikea reuna. Nämä vaihtoehdot ja niitä vastaavat valitsimet on koottu taulukkoon 3.7. Saman komennon avulla voi määrittää useitakin paikkoja, kun ne erottaa pilkulla, esimerkiksi seuraavalla tavalla:

```
\fancyhf{} % Tyhjennetään ylä- ja alätunnisteet.
\fancyhf[HEL,HOR]{\thepage}
```

Edellä mainittu komento sijoittaa sivunumeron (`\thepage`) ylätunnisteseen (H) parillisten sivujen (E) vasempaan reunaan (L) ja parittomien sivujen (O) oikeaan reunaan (R). Käytännössä siis aukeaman ulkoreunoihin ladotaan sivunumerot. Seuraava esimerkki sijoittaa ajatusviivoilla (–) reunustetun sivunumeron kaikille sivuille alätunnisteseen (F) sivun keskelle (C):

```
\fancyhf[FC]{-- \thepage\ --}
```

Paketin `fancyhdr` avulla voi myös määritellä Latexin sivutyylejä toisenlaiseksi tai luoda kokonaan omia sivutyylejä. Nämä tehdään komennolla `\fancypagestyle`, jonka ensimmäinen argumentti on sivutyylin nimi ja toinen argumentti on sivutyylin määritelmä. Määritelmä sisältää tarvittavat `\fancyhf`-komennot, joilla ylä- ja alätunnisteet määritellään.

Seuraava esimerkki tekee Latexin plain-sivutyylistä saman kuin fancy-tyylin, eli sekin noudattaa `\fancyhf`-komennolla määriteltäviä ylä- ja alätunnisteita:

```
\fancypagestyle{plain}{}
```

Seuraava esimerkki määrittelee kokonaan oman sivutyylin:

```
\fancypagestyle{omatyyl1}{  
  \fancyhf{}  
  \fancyhf[FEL, FOR]{\thepage}  
}
```

Ylä- ja alatunniste voidaan erottaa tekstipalstasta vaakasuuntaisella viivalla, jonka leveyttä on mahdollista muuttaa määrittelemällä uudelleen komennot `\headrulewidth` ja `\footrulewidth`. Komennon määritelmäksi kirjoitetaan Texin mitta. Seuraavassa esimerkissä asetetaan yhden työpografisen pisteen (1 bp) levyiset viivat. Mitan voisi asettaa myös nolaksi (0 bp), jolloin erotinviiva katoaa kokonaan näkyvistä.

```
\renewcommand{\headrulewidth}{1bp} % ylätunnisteen erotinviiva  
\renewcommand{\footrulewidth}{1bp} % alatunnisteen erotinviiva
```

Mikäli haluaa omiin ylä- ja alatunnisteisiin esimerkiksi päälukujen ja alalukujen nimiä, täytyy tunnisteisiin sisällyttää aiemmin kuvatut `\leftmark`- ja `\rightmark`-komennot. Ensinnä mainittu sisältää pääluvun nimen ja jälkimmäinen alaluvun nimen.

```
\fancyhf[HEL]{\leftmark}  
\fancyhf[HOR]{\rightmark}
```

Latex latoo pää- ja alalukujen nimet oletuksena versaalikirjaimilla eli isoilla kirjaimilla. Jos ne haluaa johonkin toiseen muotoon, täytyy itse määrittellä uudelleen komennot `\chaptermark`, `\sectionmark` tai `\subsectionmark` (vain kaksi näistä) ja käyttää määritelmässä komentoja `\markboth` ja `\markright`. Esimerkkiin 3.2 on koottu varsin kokonaisvaltainen koodi omien ylä- ja alatunnisteiden toteutukseen.

3.3 Pdf-tiedosto

Pdf-tiedostot voivat sisältää metatietoja kuten dokumentin nimen, aiheen, tekijän ja päiväyksen. Tiedostoille voi määrittää myös erilaisia

```

1 \documentclass{book}
2 \usepackage{fancyhdr}
3
4 % Päälukujen (\chapter) aloitussivu käyttää plain-sivutyylä.
5 % Tässä määritellään se uudestaan.
6 \fancypagestyle{plain}{
7   \fancyhf{}
8   \fancyhf[FC]{-- \thepage\ --}
9   \renewcommand{\headrulewidth}{0bp}
10  \renewcommand{\footrulewidth}{0bp}
11 }
12
13 % fancy-sivutyylin asetukset:
14 \fancyhf{}
15 \fancyhf[HEL]{\leftmark}
16 \fancyhf[HOR]{\rightmark}
17 \fancyhf[FC]{-- \thepage\ --}
18 \renewcommand{\headrulewidth}{1bp}
19 \renewcommand{\footrulewidth}{0bp}
20
21 \begin{document}
22
23 \pagestyle{fancy}
24
25 % Päälukujen (\chapter) yhteydessä komento \markboth{...}{...} määrittää
26 % sekä \leftmark- että \rightmark-komennot. Tässä jälkimmäinen
27 % määritellään tyhjäksi.
28 \renewcommand{\chaptermark}[1]{%
29   \markboth{\chaptername\ \thechapter: #1}{}}
30
31 % Alalukujen (\section) yhteydessä komento \markright{...} määrittää
32 % vain \rightmark-komennon.
33 \renewcommand{\sectionmark}[1]{\markright{\thesection\ #1}}

```

Esimerkki 3.2: Omien ylä- ja alatunnisteiden toteuttaminen.

```

1 \hypersetup{
2   hidelinks, bookmarksopen, bookmarksnumbered,
3   pdfinfo={
4     Title={Laatikollinen lateksia},
5     Subject={Opas lateksiin ja liisteriin},
6     Author={Lauri Liisteri}
7   }
8 }

```

Esimerkki 3.3: `\hypersetup`-komennolla asetetaan `hyperref`-paketin asetuksia, esimerkiksi pdf:n metatietoja.

asetuksia kuten ristiviitteiden ja linkkien ulkoasun tai sisäisen sisällysluettelon ominaisuuksia. Pdf-tiedoston asetukset toteutetaan `hyperref`⁷-paketin avulla. Paketti neuvotaan lataamaan muiden pakettien jälkeen, koska se lisää ominaisuuksia muihin komentoihin. Paketin voisi ladata esimerkiksi seuraavalla tavalla:

```

% Muiden \usepackage-komentojen jälkeen.
\usepackage[unicode]{hyperref}

```

Esimerkissä käytetty valitsin `unicode` aiheuttaa sen, että pdf-tiedoston sisäisissä merkkijonoissa käytetään Unicode-merkistöä ja sen UTF-8-koodausta. Ilman tätä valitsinta pdf-tiedoston sisäisen sisällysluettelon merkistö ei välttämättä näy oikein.

`hyperref`-paketin asetuksia voi määritellä valitsimien avulla `\usepackage`-komennon yhteydessä mutta myös erillisellä `\hypersetup`-komennolla. Komennolle annetaan yksi argumentti, joka sisältää pilkuilla erotettuna erilaisia valitsimia ja niiden arvoja.

Esimerkki 3.3 havainnollistaa `\hypersetup`-komennon käyttöä. Komennon argumentissa valitsin `hidelinks` saa aikaan sen, että pdf:ssä olevia linkkejä ja ristiviitteitä ei merkitä millään tavalla. Oletuksena linkit kehystetään eri väreillä riippuen linkin tyypistä. Valitsin `bookmarksopen` näyttää pdf:n sisällysluettelon kokonaan avattuna. Oletuksena alaluvut on piilotettuna, ja ne joutuu avaamaan napsauttamalla hiirellä ylemmäntasoisien otsikon avauspainiketta. Valitsin `bookmarksnumbered` näyttää pdf:n sisällysluettelossa lukujen numeroinnin (kuten

⁷ <https://www.ctan.org/pkg/hyperref>

1.1, 1.2, 2.1 tms.). Valitsimella pdfinfo määritellään pdf:n metatietoja kuten otsikko, aihe ja tekijä mutta omiakin metatietoja voi lisätä. Paljon muitakin asetuksia on olemassa, ja niistä voi lukea lisää [hyperref](#)-paketin ohjekirjasta.

Pdf-tiedoston sisäiseen sisällysluetteloon tulevat automaattisesti mukaan samat otsikot (luku 4.4.2) kuin ladottavaan sisällysluetteloonkin (luku 4.5). Pdf-tiedoston luetteloon voi kuitenkin itse lisäillä omia otsikoita, jotka eivät näy missään muualla. Se tehdään [hyperref](#)-paketin komennolla `\pdfbookmark`:

```
\pdfbookmark[taso]{Teksti}{tunniste}
```

Komennon valinnainen argumentti `taso` on kokonaisluku, joka tarkoittaa otsikon tasoa. Se on samankaltainen tasonumero kuin otsikotason taulukossa 4.5 (s. 145). Argumentti `Teksti` on pdf:n sisällysluettelomerkinnän teksti, ja `tunniste` on mikä tahansa yksilöllinen tekstimuutoksen tunniste kyseiselle luettelomerkinnälle. Tunniste ei näy missään, mutta pdf tarvitsee sisäiseen toimintaansa jonkin yksilöllisen tunnisteen.

3.4 Fontit

Fontit ja niiden asettaminen on Latexissa melko monimutkainen kokonaisuus, koska fonteilla on paljon ominaisuuksia ja niihin vaikutetaan monilla eri asetuksilla ja abstraktiotasoilla. Aika monta asiaa pitää ymmärtää, jotta voi tehokkaasti työskennellä Latexin fonttien kanssa.

Fontti jo itsessään on moniselitteinen käsite, joka vaatii typografiassa usein täsmentäviä ilmauksia. Sana *fontti* voi tarkoittaa kokonaista kirjainperhettä eli yhteensopivien kirjainleikkausten muodostamaa kokonaisuutta. Samaan kirjainperheeseen kuuluu yleensä ainakin neljä eri leikkausta: tavallinen, *kursiivi*, **lihavoitu** ja **lihavoitu kursiivi**. Joihinkin perheisiin kuuluu leikkauksia paljon enemmänkin, kuten useita eri vahvuuksia. Joissakin puheissa sana *fontti* tarkoittaa vain yhtä kirjainleikkausta, ja silloin koko perheeseen viitataan ehkä sanalla fonttiperhe.

Tässä oppaassa käytetään *fontti*-sanaa yleisnimityksenä Latexin kirjaintyypeihin liittyville asetuksille. Se tarkoittaa kirjainperhettä tai sii-

hen kuuluvaa yksittäistä leikkausta sekä asetuksia, jotka liittyvät niihin. Silloin kun merkitystä pitää täsmentää, käytetään suomenkielisiä nimiä kirjainperhe ja kirjainleikkaus. Sen sijaan sana *kirjasin* on jätetty kokonaan pois. Se tarkoittaa vanhassa metalliladonnassa ja mekaanisissa kirjoituskoneissa metallisen ladontakappaleen eli kirjakkeen päähän valettua kirjaimen tai muun merkin kohokuviota, joka painaa mustejäljen paperille.

Kuten Latexissa yleensäkin myös fonttien kanssa kannattaa käyttää korkean abstraktiotason komentoja, jotka piilottavat yksityiskohdat ja teknisen toteutuksen. Latexin fonttitoiminnot on suunniteltu juuri siihen: matalan tason fonttiasetukset määritellään mieluiten vain kerran dokumentin esittelyosassa, ja sen jälkeen käytetään pelkästään korkean tason komentoja.

Latexin fonttitekniikka rakentuu eri-ikäisistä kerroksista ja tekniikoista. Fontteja on aikoinaan tehty Metafont⁸-nimisellä kielellä, jolla kuvataan merkkien muodot. Metafont on myös tietokoneohjelma, joka tuottaa kuvauskielen perusteella bittikarttafontteja eli pikseleistä koostuvia fontteja. On käytetty myös kehittyneempää Metapost⁹-kuvauskieltä ja -ohjelmaa, joilla on tuotettu vektorigrafiikkafontteja EPS- eli Encapsulated Post Script -muodossa ja muutettu niitä edelleen Post Script Type 1 -fonteiksi. Myöhemmin mukaan ovat tulleet nykyaikaiset True Type- ja Open Type -fontit, ja niihin tämä opas keskittyy.

3.4.1 Fonttien määrittäminen

Latexin fonttien perustoiminnot rakentuvat kolmen erityyppisen kirjainperheen varaan:

1. peruskirjainperhe eli dokumentin pääasiallinen kirjainperhe, joka on kirjatypografiassa usein antiikva eli pääteviivallinen (serif, roman)¹⁰
2. groteski eli pääteviivaton (sans serif, gothic)¹¹

⁸ <https://www.ctan.org/pkg/metafont>

⁹ <https://www.ctan.org/pkg/metapost>

¹⁰ Antiikva (lat. *antiquus* 'vanha') perustuu antiikin Roomassa käytettyihin kirjainmuotoihin. Niissä on pääteviivat, ja viivojen vahvuus vaihtelee.

¹¹ Groteskiin (ransk. *grotesque* 'kummallinen') kuuluu pääteviivojen puuttumisen lisäksi lähes tasavahuiset kirjainten viivat. Tämän oppaan groteskifontissa on kuitenkin sel-

Amf Amf Amf

Kuva 3.1: Vasemmalla pääteviivallinen, keskellä pääteviivaton ja oikealla tasalevyinen pääteviivallinen kirjainperhe.

3. tasalevyinen kirjoituskoneen kaltainen perhe (typewriter, monospaced, teletype).

Kuvassa 3.1 ovat tässä oppaassa käytetyt kolme eri kirjainperhettä. Leipätekstissä käytetään antiikvaa, otsikoissa ja kuvateksteissä groteskia ja koodiesimerkeissä tasalevyistä. Kirjoituskoneen kaltainen tasalevyinen kirjainperhe on tässä tapauksessa tyypiltään antiikva eli pääteviivallinen, mutta se voisi olla muutakin. Tasalevyisyys on sen kirjainperheen tärkein määrittävä tekijä Latexin asetusten näkökulmasta.

Joidenkin fonttien käyttöönottoon on tehty oma pakettinsa, joten sellaiset fontit voi ladata dokumentin esittelyosassa komennolla `\usepackage`. Fonttikohtaisia paketteja on olemassa varsinkin vanhalle fonttitekniikalle (Metafont, Post Script Type 1) mutta myös matematiikkatilan fonttiasetuksille (luku 3.4.12) ja joillekin kirjainperhekokonaisuuksille.

Latexissa pisimmälle ”tuotteistettu” kokonaisuus taitaa olla Libertinus-kirjainperhe, joka sisältää antiikvan, groteskin ja tasalevyisen kirjainperheen sekä matematiikkatilan symboleita. Libertinus-kirjainperheet saa käyttöön lataamalla paketti `libertinus`¹²:

```
\usepackage{libertinus}
```

Valmiita paketteja on kuitenkin vain harvoille fonteille, ja käytännössä lähes aina True Type- ja Open Type -muodossa olevat fontit otetaan käyttöön `fontspec`¹³-paketin komennoilla seuraavan esimerkin mukaisesti:

```
\setmainfont{TeX Gyre Termes}[Scale=1]
\setsansfont{TeX Gyre Heros} [Scale=MatchLowercase]
\setmonofont{TeX Gyre Cursor}[Scale=MatchLowercase]
```

västi antiikvamaiset kaksivahvuiset viivat, joten se on eräänlainen antiikvan ja groteskin välimuoto.

¹² <https://www.ctan.org/pkg/libertinus>

¹³ <https://www.ctan.org/pkg/fontspec>

Edellisessä esimerkissä TeX Gyre Termes, Heros ja Cursor ovat kirjainperheiden nimiä. Fonttitiedostojen tulee olla asennettuna käyttöjärjestelmän normaalien käytäntöjen mukaisesti tai Latex-jakelun käytäntöjen mukaisesti.

Kirjainperheiden käyttöönoton yhteydessä voi määritellä lukuisia asetuksia kuten ligatuureja, gemenanumeroita, optisia kokoja ja muita fontin ominaisuuksia. Edellisessä esimerkissä käytetään vain Scale-valitsinta, jolla fontin voi skaalata haluttuun kokoon.

Peruskirjainperheen (`\setmainfont`) skaalaukseksi asetetaan esimerkissä `Scale=1`, eli sille ei tehdä mitään, ja koko valitsimen voisi jättää pois. Sen sijaan kahdella muulla kirjainperheellä (`\setsansfont`, `\setmonofont`) käytetään kerroinasetusta `MatchLowercase`, joka skaalaa fontin siten, että gemenakirjaimet eli pienet kirjaimet ovat yhtä korkeita kuin peruskirjainperheessä. Mikäli skaalausasetus `MatchLowercase` ei tuota ihan toivottua tulosta, voi kirjainperheen skaalausta hienosäätää vielä `ScaleAgain`-valitsimella seuraavalla tavalla:

```
\setmonofont{TeX Gyre Cursor}
[Scale=MatchLowercase, ScaleAgain=.97]
```

Kirjainperheen määrittelyn yhteydessä ei yleensä tarvitse antaa kuin kirjainperheen nimi, sillä `fontspec`-paketti ja kääntäjät osaavat automaattisesti ladata perheeseen sisältyviä eri fonttitiedostoja kuten pystyasentoisen leikkauksen, kursiivin ja lihavoinnin. On kuitenkin mahdollista määritellä kirjainperheeseen kuuluvia leikkausten nimiä tai fonttitiedostoja erikseen. Tällainen on tarpeen esimerkiksi silloin, kun kirjainperhe sisältää useita eri vahvuuksia ja halutaan itse määritellä, mikä niistä tulee perusvahvuudeksi ja mikä lihavoiduksi.

Esimerkki 3.4 selventää, kuinka kirjainperheen eri leikkausten nimet tai fonttitiedostot määritellään. Kullekin leikkaukselle voi määrittää myös omat asetuksensa `Features`-sanaan päättyvällä valitsimella. Samaa asiaa havainnollistetaan myös konkreettisemmin esimerkissä 3.5. Leikkauksen nimessä voi käyttää tähteä (*), joka korvautuu kirjainperheen nimellä (`Macklin Text`).

Jos edellä kuvatut kolme kirjainperhettä (`\setmainfont`, `\setsansfont` ja `\setmonofont`) eivät riitä, on `fontspec`-paketissa komennot lisäperhei-


```

1 \setmainfont {...} [
2   UprightFont={...},      UprightFeatures={...},
3   ItalicFont={...},      ItalicFeatures={...},
4   BoldFont={...},        BoldFeatures={...},
5   BoldItalicFont={...},  BoldItalicFeatures={...},
6   SlantedFont={...},     SlantedFeatures={...},
7   BoldSlantedFont={...}, BoldSlantedFeatures={...},
8   SmallCapsFont={...},   SmallCapsFeatures={...}]

```

Esimerkki 3.4: Kirjainperheeseen sisältyvien leikkausten nimien ja kirjainleikkauskohtaisten ominaisuuksien määrittely.

```

1 \setmainfont{Macklin Text}[
2   UprightFont   = {* Light},
3   ItalicFont    = {* Light Italic},
4   BoldFont      = {* Medium},
5   BoldItalicFont = {* Medium Italic}]

```

Esimerkki 3.5: Eri leikkausten nimien määrittely Macklin Text -kirjainperheelle. Leikkauksen nimessä tähti (*) korvautuu automaattisesti koko perheen nimellä.

den ja -leikkausten määrittämiseen. Uusi perhe määritellään seuraavasti:

```
\newfontfamily{\hienoperhe}{TeX Gyre Schola}[...]
```

Komento `\newfontfamily` toimii samalla tavalla kuin aiemmin esitellyt `\setmainfont` ym. komennot, mutta lisäksi ensimmäisellä argumentilla nimetään komento, jolla kirjainperhe otetaan käyttöön. Edellisessä esimerkissä luodaan komento `\hienoperhe`, joka kytkee päälle TeX Gyre Schola -nimisen kirjainperheen.

Jos ei tarvita kokonaista perhettä vaan yksi leikkaus riittää, käytetään komentoa `\newfontface`. Seuraavassa esimerkissä määriteltävä komento `\hienoleikkaus` ottaa käyttöön lihavoidun (bold) kirjainleikkauksen perheestä TeX Gyre Schola.

```
\newfontface{\hienoleikkaus}{TeX Gyre Schola Bold}[...]
```

3.4.2 Fontin koko ja rivikorkeus

Fonttien koot on tapana valita ja ilmaista typografisen pistemitan avulla. Esimerkiksi 10–12 pistettä on tyypillinen leipätekstin oletuskoko

tekstinkäsittelyohjelmissa. Piste on typografiassa mittayksikkö, jonka pituus on määritelty eri tavoin eri aikoina ja eri kulttuureissa.

Myös Latexissa fonttien koot voi määritellä pistemittojen avulla. Niitä ja muitakin Latexin mittayksiköitä käsitellään tarkemmin luvussa 2.4. Fonteissa oletusmittayksikkönä on vanha pica-järjestelmän piste, jonka pituus on noin 0,3515 millimetriä. Sen lyhenne Latexissa on pt. Tämän oppaan esimerkeissä käytetään kuitenkin Post Script -standardin mukaista, julkaisuohjelmiin vakiintunutta uudempaa pica-pistettä, joka on hieman edellistä pidempi: noin 0,3528 millimetriä. Latexissa sen lyhenne on bp. Ero näiden kahden pistemitan välillä on hyvin pieni, tavallisilla fonttiko'illa käytännössä merkityksetön.

Kirjainleikkauksen koko mitataan merkistön ylimmän ja alimman kohdan välillä, esimerkiksi k-kirjaimen ylimmän pisteen ja y-kirjaimen alimman pisteen välillä. Lisäksi mitaan luetaan mukaan merkistön ylä- ja alapuolella oleva pieni tyhjä tila, jonka fontin suunnittelija on määritellyt.

Matalalla tasolla fonttien kokoon vaikuttaa Latexissa eräs yllättävä asia. Nimittäin dokumenttiluokalle (luku 3.1) voi antaa valitsimen, jolla fontin koko asetetaan. Vaihtoehtoja on Latexin normaaleissa dokumenttiluokissa vain kolme: 10pt (oletus), 11pt ja 12pt. Dokumenttiluokan kokoasetus vaikuttaa myös sivun marginaaleihin, koska Latex pyrkii pitämään rivin merkkimäärän lukijalle sopivana: yhdelle riville ei kannata latoa ihan mahdottomasti merkkejä, koska kovin pitkän rivin seuraaminen rasittaa lukijaa ja vaatii enemmän keskittymistä.

Fontin koon määrittäminen dokumenttiluokan valitsimella ehkä kuuluu jo vähän menneisyyteen, mutta voi sitä edelleen käyttää, jos se riittää ja sillä saa halutun lopputuloksen. Yleensä lienee järkevää jättää dokumenttiluokan fonttiasetus oletukseksi (10pt) ja käyttää koon asettamiseen luvuissa 3.4.4 ja 3.4.5 kerrottuja tapoja. Sivun marginaalien ja muiden mittojen määrittämiseen on ohjeita luvussa 3.2.

Fonttiasetukseen kuuluu fontin koon lisäksi toinenkin mitta: rivikorkeus (`\baselineskip`). Se on peräkkäisten rivien peruslinjojen välinen etäisyys. Fontin koko ja rivikorkeus määritellään samanaikaisesti, koska ne ovat saman `\fontsize`-komennon argumentteja. Esimerkki:

```
\fontsize{10bp}{12bp} \selectfont
```

```

1 \fontsize{8bp}{11bp}\selectfont Tässä on pienehkö leipätekstin
2 fonttikoko ja suhteellisen suuri rivikorkeus. Pitkät rivit vaativat
3 suuremman rivikorkeuden kuin lyhyet rivit.
4
5 \fontsize{16bp}{17bp}\selectfont Tässä on melko suuri fontti ja
6 suhteellisen pieni rivikorkeus. Suuri fontti ja lyhyet rivit eivät
7 tarvitse kovin suurta rivikorkeutta.

```

⇒ Tässä on pienehkö leipätekstin fonttikoko ja suhteellisen suuri rivikorkeus. Pitkät rivit vaativat suuremman rivikorkeuden kuin lyhyet rivit.

Tässä on melko suuri fontti ja suhteellisen pieni rivikorkeus. Suuri fontti ja lyhyet rivit eivät tarvitse kovin suurta rivikorkeutta.

Esimerkki 3.6: Fontin koon ja rivikorkeuden asettaminen ja vaikutus.

Ensimmäinen argumentti on fontin kokomitta ja toinen on rivikorkeus. Mittayksiköt voivat olla mitä tahansa Texin mittoja, ja oletuksena käytetään pt-pistemittaa, jos yksikköä ei ole mainittu. Komento `\selectfont` on mukana, koska vasta sen myötä matalan tason fonttikomennot tulevat voimaan. Korkean tason fonttikomennot (luku 3.4.3) suorittavat sen automaattisesti.

Rivikorkeus on vähintään sama kuin fontin koko, mutta yleensä se asetetaan pari pistettä suuremmaksi, jotta rivit eivät olisi liian lähellä toisiaan. Esimerkissä 3.6 on kaksi erilaista `\fontsize`-komentoa ja la-dotu lopputulos.

Toinen tekstirivien peruslinjojen väliseen etäisyyteen vaikuttava asetus on `\baselinestretch`. Se on desimaalilukukerroin, jolla nykyinen rivikorkeus kerrotaan. Kerroin asetetaan helpoimmin komennolla `\linespread`.¹⁴

```
\fontsize{10bp}{12bp} \linespread{1.3} \selectfont
```

Edellä oleva esimerkki asettaa fontin kooksiksi 10 pistettä ja rivikorkeudeksi 12 pistettä. `\linespread`-komennolla asetetun kertoimen vuoksi rivien peruslinjojen väliseksi etäisyydeksi tulee lopulta 1,3 kertaa 12

¹⁴ Toinen tapa: `\renewcommand{\baselinestretch}{kerroin}`

Komento		Merkitys	
<code>\rmfamily</code>	<code>\textrm{...}</code>	perhe	perus, yl. antiikva, serif, roman
<code>\sffamily</code>	<code>\textsf{...}</code>	perhe	groteski, sans serif, gothic
<code>\ttfamily</code>	<code>\texttt{...}</code>	perhe	tasalevyinen, typewriter
<code>\mdseries</code>	<code>\textmd{...}</code>	leikkaus	tavallinen vahvuus, medium
<code>\bfseries</code>	<code>\textbf{...}</code>	leikkaus	lihavoitu, bold
<code>\upshape</code>	<code>\textup{...}</code>	leikkaus	pystyasento, tavallinen
<code>\itshape</code>	<code>\textit{...}</code>	<i>leikkaus</i>	kursiivi, italic
<code>\slshape</code>	<code>\textsl{...}</code>	<i>leikkaus</i>	kalteva, slanted, oblique
<code>\scshape</code>	<code>\textsc{...}</code>	LEIKKAUS	pienversaali, kapiteeli, small caps

Taulukko 3.8: Komennot kirjainperheen ja kirjainleikkauksen valintaan. Perustila on `\rmfamily`, `\mdseries` ja `\upshape`.

pistettä eli 15,6 pistettä. Ei ole väliä, kummassa järjestyksessä `\font-size-` ja `\linespread-` komennot annetaan. Asetukset tulevat voimaan vasta `\selectfont`-komennon jälkeen.

Käytännössä `\linespread` sopii rivikorkeuden yleistason hienosäätöön esimerkiksi dokumentin esittelyosassa. Tilannekohtainen rivikorkeus on parasta asettaa `\fontsize`-komennolla.

3.4.3 Kirjainperheen ja -leikkauksen valitseminen

Latexissa on joukko korkean tason fonttikomennoja, jotka on tarkoitettu käytettäväksi sen jälkeen, kun matalan tason asetukset on kerran määritetty. Taulukossa 3.8 on komennot kirjainperheen ja kirjainleikkauksen valintaan. Joka rivillä ensin mainittu komento (esim. `\rmfamily`) vaikuttaa tekstiin, joka tulee komennon jälkeen. Vaikutusalue rajoittuu nykyisen ympäristön (luku 2.3) sisään tai aaltosulkeilla (luku 2.1.6) rajatun alueen sisään. Rivillä toisena olevalle komennolle (esim. `\textrm`) annetaan yksi argumentti, ja komennon vaikutus koskee vain argumenttina olevaa tekstiä.

Taulukossa on ensimmäisillä riveillä kirjainperhekomennot (3 kpl), jotka vaihtavat koko perheen kaikkine leikkauksineen. Niiden jälkeen tulevat kirjainleikkauksen komennot, jotka valitsevat toisen leikkauksen samasta perheestä. Kirjainleikkauksen asetukset jaetaan kahteen ryhmään: *series* 'sarja' ja *shape* 'muoto'. Kummastakin ryhmästä on va-

Komento	10pt	11pt	12pt
<code>\tiny</code>	5	6	6
<code>\scriptsize</code>	7	8	8
<code>\footnotesize</code>	8	9	10
<code>\small</code>	9	10	10,95
<code>\normalsize</code>	10	10,95	12
<code>\large</code>	12	12	14,4
<code>\Large</code>	14,4	14,4	17,28
<code>\LARGE</code>	17,28	17,28	20,74
<code>\huge</code>	20,74	20,74	24,88
<code>\Huge</code>	24,88	24,88	24,88

Taulukko 3.9: Fonttien oletuspistekoot dokumenttiluokkien valitsimilla 10pt, 11pt ja 12pt.

littuna aina yksi ominaisuus, eli samanaikaisesti voi olla voimassa esimerkiksi `\bfseries` (**lihavoitu**) ja `\itshape` (*kursiivi*), ja tuloksena on ***lihavoitua kursiiivia***.

Useimmissa fonteissa kursiiivileikkaus (`\itshape`) ja kalteva leikkaus (`\slshape`) tuottavat saman lopputuloksen, mutta käsitteellisesti ne ovat eri asia. Kursiivi on aina muodoltaan erilainen leikkaus, joka hieman mukaillee käsialakirjoitusta, joskaan kirjaimia ei ole sidottu toisiinsa. Sen sijaan kalteva leikkaus on tavallisen eli pystyasentoisien leikkauksen kallistettu versio. Kallistus voi olla kirjainmuotoilijan piirtämä leikkaus, tai kallistus on voitu toteuttaa mekaanisesti. Kallistuksen voi tehdä myös Latexissa (luku 3.4.10).

Komennot fontin koon valintaan ovat taulukossa 3.9. Taulukko kertoo myös, mitä fontin pistekokoa (pt) mikäkin komento tarkoittaa oletuksena. Oletus riippuu Latexin dokumenttiluokkien (luku 3.1.1) fonttikokovalitsimista 10pt, 11pt ja 12pt.

Kaikille korkean tason fonttikomennoille on olemassa myös saman niminen ympäristönsä, esimerkiksi `rmfamily`, `bfseries`, `itshape` tai `small`. Seuraavassa esimerkissä on kaksi fontteihin vaikuttavaa ympäristöä sisäkkäin.

```

1 \documentclass[10pt]{article} % 10pt on oletus
2 \usepackage{fontspec}
3
4 \setmainfont{TeX Gyre Termes}[Scale=1.4]
5 \setsansfont{TeX Gyre Heros} [Scale=MatchLowercase]
6 \setmonofont{TeX Gyre Cursor}[Scale=MatchLowercase]
7 \linespread{1.45}

```

Esimerkki 3.7: Fonttikokojen määrittäminen suhteessa peruskirjainperheeseen.

```

\begin{footnotesize}
  \begin{scshape}
    Tämä teksti on pientä pienversaalia.
  \end{scshape}
\end{footnotesize}

```

⇒ TÄMÄ TEKSTI ON PIENTÄ PIENVERSAALIA.

3.4.4 Fonttikoon määrittely suhteellisesti

Dokumentin fonttien koot on helpointa määrittää siten, että asettaa ensin peruskirjainperheen koon ja antaa muiden fonttien määräytyä suhteessa siihen. Esimerkki 3,7 selventää, kuinka se tapahtuu. Alussa otetaan käyttöön dokumenttiluokka `article` ja annetaan sille valitsin `10pt`, joka määrittää fonttikooksi 10 pistettä. Se on dokumenttiluokan oletusasetus, jota ei tarvitsisi edes kirjoittaa näkyviin. Esimerkin toisella rivillä otetaan `fontspec`-paketti käyttöön.

Peruskirjainperheen (rivi 4) koko skaalataan 1,4-kertaiseksi, eli pistekooksi tulee 1,4 kertaa 10 pistettä eli 14 pistettä (pt). Normaalikokoinen peruskirjainperhe on ainoa, jonka pistekoko tiedetään. Kaikkien muiden koot täytyisi selvittää laskemalla.

Groteski eli pääteviivaton kirjainperhe (rivi 5) ja tasalevyinen perhe (rivi 6) skaalataan samankorkuiseksi kuin perusperhe. Vertailukohdana ovat gemenat eli pienaakkoset (`MatchLowercase`). Näiden kahden kirjainperheen pistekokoa ei tiedetä. Se ei välttämättä ole sama kuin perusfontissa, koska fonttien pistekoko mitataan ylimmän ja alimman kohdan välillä ja koska fonttien mittasuhteet ovat erilaisia.

Viimeisellä rivillä oleva `\linespread`-komento on tärkeä. Se asettaa rivikorkeuden kertoimeksi 1,45. Kertoimen täytyy olla vähintään yhtä

suuri kuin peruskirjainperheen skaalauskerroin (1,4), jotta rivivälit ovat riittävän suuret. Näiden asetusten jälkeen dokumentissa käytetään korkeamman tason komentoja fonttien valintaan, esimerkiksi fonttikoon valintakomentoja `\small`, `\normalsize`, `\large` (taulukko 3.9).

Edellä kuvatussa suhteellisessa kirjainperheiden koon määrittelyssä on sellainen ongelma tai kummallisuus, että Latex koko ajan luulee, että peruskirjainperhe on normaalikokoisena 10 pistettä (pt). Latexin matalan tason fonttikomennot eivät tiedä kirjainperheen skaalauskerroimesta, ja siksi esimerkiksi komentojen

```
\fontsize{10bp}{12bp} \selectfont
```

tuloksena ei todellisuudessa ole 10 pisteen (bp) fontti, vaan mukaan lasketaan myös kirjainperheen skaalauskerroin. Tämän vuoksi `\fontsize`-komennon käyttö menee aika oudoksi. Argumenttina annettu kokomitta ei pidä paikkaansa.

Jos korkean tason fonttikokokomentojen (taulukko 3.9) lisäksi tarvitaan jotakin muuta kokoa, voisi mahdollisesti `\fontsize`-komennon sijasta käyttää `fontspec`-paketin tarjoamaa komentoa ja tilanteeseen sopivaa skaalauskerrointa esimerkiksi seuraavalla tavalla:

```
{\addfontfeatures{Scale=3.2} Poikkeuksellisen isoa tekstiä}
```

Jos edellä mainitut kummallisuudet eivät häiritse eikä ole tarvetta määrittellä fontteja tarkasti tietyn pistekoon mukaiseksi, on suhteellinen määrittelytapa todella helppo. Kaikki dokumentin fontit määräytyvät perusfontin skaalaukertoimen kautta. Tämä tapa sopii hyvin varsinkin dokumentin sisällön kirjoittamisvaiheeseen, jossa ehkä halutaan vain nopeasti asettaa dokumentti suurin piirtein järkevän näköiseksi. Myöhemmin voi määrittää koot tarkemmin niin sanotun absoluuttisen menetelmän avulla, jota käsitellään seuraavassa alaluvussa.

3.4.5 Fonttikoon määrittely absoluuttisesti

Absoluuttinen fonttien koonmäärittelytapa tarkoittaa sitä, että koot asetetaan tietyn kokoiseksi käyttämällä esimerkiksi pistemittoja ja että kirjaimet myös päättyvät lopulliseen dokumenttiin juuri sen kokoisena. Tämä tapa on myös teknisesti eheä, eli Latexin eri osat ovat samaa mieltä

```

1 \documentclass{article}
2 \usepackage{fontspec}
3
4 % Leipätekstiin samankokoiset fontit
5 \setmainfont{TeX Gyre Termes}
6 \setsansfont{TeX Gyre Heros} [Scale=MatchLowercase]
7 \setmonofont{TeX Gyre Cursor}[Scale=MatchLowercase]
8
9 % Muualle sans ja mono ilman skaalausta
10 \newfontfamily{\sffamilyabs}{TeX Gyre Heros}
11 \newfontfamily{\ttfamilyabs}{TeX Gyre Cursor}
12
13 \linespread{1} % ei välttämättä tarvita
14
15 % Kaikki tarvittavat fonttikoot ja komennot
16 \renewcommand{\footnotesize}{\fontsize{10bp}{12bp}\selectfont}
17 \renewcommand{\small}      {\fontsize{12bp}{14bp}\selectfont}
18 \renewcommand{\normalsize} {\fontsize{14bp}{17bp}\selectfont}
19 \renewcommand{\large}      {\fontsize{17bp}{19bp}\selectfont}
20 \renewcommand{\Large}      {\fontsize{20bp}{22bp}\selectfont}
21
22 \normalsize % jotta tulee heti voimaan eikä vasta tekstiosassa

```

Esimerkki 3.8: Fonttikokojen määrittäminen pistekoon avulla.

siitä, minkäkokoisesta fontista on kyse. Näin ei ollut suhteellisen tavan kanssa (luku 3.4.4).

Joskus oppilaitoksen, yhtiön tai muun julkaisijan ohjeissa määritellään tarkasti, mitä fontteja käytetään ja mikä on leipätekstin ja otsikoiden fonttikoko. Silloin tarvitaan tässä luvussa kuvattua tapaa fonttien asettamiseen.

Esimerkistä 3.8 selvää perusajatus. Peruskirjainperhe (rivi 5) otetaan käyttöön ilman skaalausta (`Scale=1`), minkä vuoksi koon voi jatkossa asettaa täsmälleen kohdalleen `\fontsize`-komennolla. Samaa ei tehdä groteskin eikä tasalevyisen fontin kanssa (rivit 6–7), vaan käytetään skaalausta `MatchLowercase`, jotta tekstikappaleessa kaikki kirjainperheet näyttävät samankokoisilta. Tässä menetetään mahdollisuus määrittää näiden kirjainperheiden koko täsmällisesti pistemitan avulla. Jos siihen on tarvetta esimerkiksi otsikoissa, voidaan käyttää rivien 10–11

komentoja. Niillä luodaan uudet kirjainperheet, jotka ovat käytännössä samoja mutta ilman skaalausta.

Uusien skaalaamattomien kirjainperheiden kommentojen nimiksi on valittu `\sffamilyabs` ja `\ttfamilyabs` (vrt. `\sffamily` ja `\ttfamily`, taulukko 3.8), ja näillä komennoilla kirjainperheet kytketään päälle. Jos esimerkiksi jonkin julkaisun vaatimuksiin kuuluu, että otsikossa täytyy olla 20 pisteen lihavoitu TeX Gyre Heros -kirjainleikkaus, voi esimerkiksi 3.8 olevien asetusten pohjalta antaa otsikon ulkoasun (luku 4.4.4) määrittelyn yhteydessä seuraavat komennot:

```
\sffamilyabs\Large\bfseries
```

Esimerkin 3.8 riveillä 16–20 määritellään uudelleen Latexin korkean tason komennot, joilla fonttikoot asetetaan. Oletusarvot tulevat dokumenttiluokasta (luku 3.1), mutta jos ne eivät ole sopivia, täytyy vähintäänkin määritellä komento `\normalsize` mutta sen lisäksi kaikki ne koot, joita omassa dokumentissa tarvitaan. Tässä esimerkissä normaali koko asetetaan 14 pisteen kokoiseksi.

Jokaiselle fonttikoolle määritetään riveillä 16–20 myös oma rivikorkeus, ja se on tarkoitus asettaa sopivaksi juuri kyseiselle koolle. Rivikorkeuteen vaikuttaa myös kerroin `\baselinestretch`, joka asetetaan komennolla `\linespread`. Sitä ei välttämättä tarvitse käyttää, koska kirjainperheitä ei ole skaalattu ja koska rivikorkeus asetetaan aina `\font-size`-komennolla. `\linespread` on kuitenkin kätevä komento rivikorkeuden säätämiseen yleisesti kaikkialla.

Fonttikokojen määrittelyn lopuksi rivillä 22 suoritetaan komento `\normalsize`, jotta se tulee heti voimaan. Dokumentin esittelyosassa voidaan käyttää fonttikokoon viittaavia mittayksiköitä `em` ja `ex`, ja ne viittaavat nyt tähän kokoon. `\normalsize`-komento suoritetaan kylä myöhemmin automaattisesti dokumentin tekstiosan eli `document`-ympäristön alussa.

Edellä kuvatun absoluuttisen koonmäärittystavan etuna on se, että kirjoittaja hallitsee fonttien kokoa ja rivikorkeuksia tarkasti ja että julkaisuun saadaan juuri ne mitat, jotka halutaan tai vaaditaan. Tapa on myös teknisesti eheä eli toimii Latexin sisäisen logiikan näkökulmasta

oikein. Haittana voi pitää sitä, että kaikki koot täytyy määritellä erikseen.

3.4.6 Fonttien oletusasetuksia

Oletuksena Latex-dokumentin peruskirjainperheessä (`\setmainfont`) ja pääteviivattomassa kirjainperheessä (`\setsansfont`) ovat päällä muun muassa seuraavat asetukset:

```
Ligatures={TeX, Common}
HyphenChar=-
```

TeX-ligatuurit tarkoittavat lainausmerkkien ja ajatusviivojen tuottamiseen tarkoitettuja Texin merkintätapoja kuten ' ' ja --, joita käsitellään luvuissa 2.1.9 ja 2.1.10. Ne eivät ole varsinaisia luonnollisen kielen eivätkä typografisia ligatuureja vaan kuuluvat ainoastaan Tex-kielen merkintätapoihin. Common-ligatuurit sen sijaan ovat oikeita typografisia ligatuureja kuten fi, ff ja fl, ja niitä käsitellään tarkemmin luvussa 3.4.7. Molemmat edellä mainitut ligatuurityypit saa pois päältä seuraavalla asetuksella:

```
Ligatures={TeXReset, NoCommon}
```

Valitsin HyphenChar asettaa tavutusmerkin kyseiselle kirjainperheelle. Oletuksena se on yhdysmerkki (-), mutta merkin vaihtaminen toimii vain Xelatex-kääntäjän kanssa, ei esimerkiksi Lualatexilla. Sen sijaan kaikilla kääntäjillä tavutuksen voi kytkeä kirjainperheeltä pois, kun antaa arvoksi None. Tavutusta käsitellään perusteellisemmin luvussa 3.6.

Tasalevyinen kirjainperhe (`\setmonofont`) toimii oletuksena hieman eri tavalla. Niissä muun muassa ei ole lainausmerkkien ja ajatusviivojen kirjoittamiseen tarkoitettuja Tex-ligatuurit päällä. Oletusasetuksia ovat muun muassa seuraavat:

```
Ligatures=Common
HyphenChar=None
```

Yleensä tasalevyisestä kirjainperheestä kannattaa kytkeä typografiset ligatuurit pois päältä asetuksella `Ligatures=NoCommon`. Tasalevyisen fontin ajatukseen nimenomaan kuuluu, että merkit ovat samanleveyisiä eikä yhden merkin tilaan sovi sulloa useampaa kirjainta. Kaikki tasalevyiset kirjainperheet eivät edes sisällä ligatuurimerkkejä, joten ominaisuus

näyttää olevan automaattisesti pois päältä. Varmuuden vuoksi on kuitenkin hyvä lisätä asetus `Ligatures=NoCommon` tasalevyiselle perheelle.

Oletuksena tavutus on kytketty pois päältä tasalevyiseltä kirjainperheeltä, koska sitä käytetään tavallisesti tietokoneisiin liittyvien koodien tai vastaavien ilmausten latomiseen, eikä niitä haluta yleensä tavuttaa. Tavutuksen saa kuitenkin päälle Xelatexissa määrittämällä tavutusmerkiksi esimerkiksi yhdysmerkin (`HyphenChar=-`) tai kaikilla kääntäjillä poistamalla tasalevyisen kirjainperheen oletusasetukset ennen sen määrittelyä seuraavasti:

```
\defaultfontfeatures[\ttfamily]{}  
\setmonofont{...}[...]
```

Komennolla `\defaultfontfeatures` voi asettaa (joidenkin) kirjainperheiden oletusasetukset, kun komennon suorittaa ennen kirjainperheiden määrittelyä.

```
\defaultfontfeatures[\rmfamily,\sffamily]{Ligatures={TeX, Common},  
Numbers=Lowercase}
```

Komennon valinnaisella argumentilla voi rajata, mitä kirjainperheitä oletusasetukset koskevat. Valinnaiseen argumenttiin kirjoitetaan yksi tai useampia pilkulla erotettuja komentoja, joilla kirjainperheet kytketään päälle: `\rmfamily`, `\sffamily`, `\ttfamily` (taulukko 3.8, s. 84) tai komennolla `\newfontfamily` määritelty fonttikomento (luku 3.4.1).

3.4.7 Typografiset ligatuurit

Typografiset ligatuurit ovat fontissa olevia yhdistelmämerkkejä, joissa on typografisista syistä yhdistetty kaksi tai useampia kirjaimia yhteen merkkiin.¹⁵ Tällaisia ligatuureja käytettiin jo metalliladonnassa, eli massassa metallikirjakkeessa saattoi olla enemmän kuin yksi kirjain.

Typografisten ligatuurien tarkoituksena on tuottaa tyylikkäampi lopputulos kuin saataisiin erillisten kirjainten avulla. Syynä on esimerkiksi se, että erilliset peräkkäiset kirjaimet eivät ole aina keskenään

¹⁵ Typografisten ligatuurien lisäksi on olemassa myös luonnollisten kielten ligatuureja, jotka muodostavat kieleen kuuluvan kirjaimen, kuten `æ` norjan kielessä. Niillä on jokin luonnolliseen kieleen liittyvä merkitys, eikä eri merkkejä ei ole yhdistetty typografisista syistä.

fi ff ffi fl ffl fj ffj Th st ct

Kuva 3.2: Tavalliset f-alkuiset typografiset ligatuurit, vähän harvinaisempi Th sekä historialliset ligatuurit st ja ct.

yhteensopivia. Kirjainten välistys voi olla ongelmallista: sopivalle etäisyydelle asetettaessa kirjainten osat voivat mennä rumannäköisesti päällekkäin. Riippuu fontista, mitkä ligatuurit ovat tarpeellisia ja mitkä ligatuurimerkit on ylipäättäen toteutettu fontin merkistöön. Tavallisia antiikvojen ligatuureja ovat fi, ff, ffi, fl, ffl, fj ja ffj, mutta jotkin fontit sisältävät muitakin. Kuvaan 3.2 on koottu esimerkkejä.

Latex hoitaa tavallisten typografisten ligatuurien latomisen automaattisesti, eli lähdedokumenttiin ei kannata kirjoittaa Unicode-merkistön typografisia ligatuurimerkkejä (esimerkiksi U+FB01 LATIN SMALL LIGATURE FI) vaan ihan tavallisia erillisiä kirjaimia. Open Type -fontin suunnittelija on päättänyt meidän puolestamme, mitkä kirjainyhdistelmät on parasta latoa ligatuurin avulla, ja oletusasetuksilla Latex noudattaa niitä ohjeita.

Jos kuitenkin yksittäisen ligatuurin muodostumisen haluaa estää, voi kirjainten väliin kirjoittaa komennon `\textcompwordmark`, joka tekee näkymättömän, juuri tähän tarkoitukseen olevan merkin U+200C ZERO WIDTH NON-JOINER.

```
fi f\textcompwordmark i
```

⇒ fi fi

Mikäli tavallisia ligatuureja ei halua käyttöön lainkaan, on parasta valita jo kirjainperheen tai -leikkauksen käyttöönotossa asetus `Ligatures=NoCommon`. Väliaikaisesti fontin asetuksiin voi vaikuttaa komennolla `\addfontfeatures`, joka sekin on `fontspec`-pakettiin sisältyvä komento.

```
{\addfontfeatures{Ligatures=NoCommon} fi fl}
```

Tavallisten ligatuurien lisäksi Open Type -fontit voivat sisältää myös harvinaisempia ligatuureja, jotka täytyy erikseen kytkeä päälle. Niiden tarkoituksena on esimerkiksi erityinen koristeellisuus tai historiallisen

kirjoitustavan jäljittely. Harvinaisia ligatuureja ei ole tarkoitus käyttää joka tilanteessa vaan ainoastaan erityisestä syystä. Kuvassa 3.2 ja tässäkin tekstikappaleessa olevat `st-` ja `ct-`ligatuurit on saatu fontin asetuksella `Ligatures=Historic`. Joissakin fonteissa sama tehdään asetuksella `Ligatures=Discretionary`. Nämä ligatuurit sopivat niin sanotun humanistisen antiikvaperheen (mm. Adobe Jenson) ja renessanssityylisen sisällön kanssa käytettäväksi.

Teknisesti on mahdollista kytkeä päälle useitakin ligatuurityyppejä samanaikaisesti:

```
Ligatures={TeX, Required, Common, Historic, Discretionary,  
Contextual}
```

Tuskin mikään fontti sisältää kaikkia ligatuurityyppejä, eikä se ole tarkoitukseen. Jotkin ligatuurit kuuluvat vain tiettyyn typografian aika-kauteen tai kirjaintyyliin. Open Type -fontin ominaisuuksia voi selvittää käyttöjärjestelmän komentotulkissa komennolla `otfinfo`. Komenton argumentiksi annetaan muun muassa fonttitiedoston nimi.

3.4.8 Numeroiden muoto

Varsinkin kirjatyypografiassa on tavallista käyttää pääasiassa gemenanumeroita (1967) eikä versaalinumeroita (1967), koska gemenanumerot sopivat leipätekstin gemenakirjainten kanssa paremmin yhteen. Versaalinumeroit puolestaan erottuvat paremmin ja sopivat esimerkiksi taulukoihin, joissa on paljon lukuja. Näihin asetuksiin vaikutetaan fontin asetuksella `Numbers`. Tosin kaikissa fonteissa ei gemenanumeroita edes ole.

```
Numbers=Lowercase % gemenanumerot  
Numbers=Uppercase % versaalinumeroit (oletus)
```

Jotkin koodi-ilmaukset sisältävät versaalikirjaimia ja numeroita sekaisin. Gemenanumerot eivät kuitenkaan sovi yhteen versaalikirjainten kanssa, koska merkkien kokoero on häiritsevän suuri. Ei siis näin: `RJ45`, `R2D2`. Gemenanumeroiden kanssa täytyy käyttää pienversaalia eli kapiteelia: `RJ45`, `R2D2`. Toinen vaihtoehto on käyttää pelkästään versaaleja: `RJ45`, `R2D2`.

Silloin kun käytetään numeroita teknisten koodien ilmaisemiseen, voi olla tarpeen merkitä numero nolla poikkiviivalla (0), jottei se sekoi- tu O-kirjaimen. Nollaan saadaan poikkiviiva seuraavalla asetuksella, jos vain fontissa on tämä ominaisuus:

```
Numbers=SlashedZero
```

Kun numeroita ladotaan taulukkoon, voidaan haluta käyttää tasalevyi- siä numeroita, jotta ne sijoittuvat allekkain samalle linjalle. Joissakin fonteissa on tarjolla tavallisten vaihtelevan levyisten numeroiden li- säksi myös tasalevyiset. Numeroiden leveyteen vaikutetaan seuraavilla asetuksilla:

```
Numbers=Monospaced % tasalevyiset numerot  
Numbers=Proportional % vaihtelevan levyiset numerot (oletus)
```

3.4.9 Välistykset: harvennus ja tiivistys

Fontin kanssa käytettäviä sanavälejä voi säätää `WordSpace`-valitsimella, jonka arvoksi annetaan desimaalilukukerroin eli suhdeluku normaaliin verrattuna. Arvona voi olla myös kolme eri kerrointa, jolloin muka- na on lisäksi sanavälin venymisen rajat. Tämä asetus on tehtävä kir- jainperheen tai -leikkauksen määrittelyn yhteydessä, eli se ei toimi esi- merkiksi `\addfontfeatures`-komennon kanssa. Katso tietoa sanaväleistä myös luvusta 2.1.2 tai tekstikappaleiden latomista käsittelevästä luvusta 4.1.

Fonttiasetusten valitsimella `LetterSpace` säädetään merkkien vä- listystä eli niiden välistä tyhjää tilaa. Arvoksi annetaan positiivinen tai negatiivinen prosenttiluku, joka ilmaisee alkuperäiseen lisättävän osuuden. Asetus `LetterSpace=6` tuottaa siis kuusi prosenttia leveämmät merkkien välit. Välistyksen tiivistäminen ja harvennuksen typografiaa käsitellään luvussa 4.2.4, mutta seuraavassa on esimerkki, kuinka ne teknisesti toteutetaan:

```
\scshape {\addfontfeatures{LetterSpace=-2} tiivistys} \\  
normaali \ \ {\addfontfeatures{LetterSpace=6} harvennus}
```

```
⇒ TIIVISTYS  
NORMAALI  
HARVENNUS
```

Jos esimerkiksi harvennuksen haluaa automaattisesti mukaan kirjainperheeseen kuuluvaan pienversaaliiin, käytetään valitsinta `SmallCapsFeatures`:

```
\setmainfont {...}[SmallCapsFeatures={LetterSpace=6}]
```

3.4.10 Keinotekoinen venytys, lihavointi ja kallistus

Valitsimella `FakeStretch` voi venyttää tai kutistaa merkkejä leveys suunnassa. Arvoksi annetaan desimaalilukukerroin, joka on suhdeluku alkuperäiseen leveyteen nähden. Seuraavassa havainnollistava esimerkki:

```
esimerkki \\  
{\addfontfeatures{FakeStretch=1.3} esimerkki}
```

⇒ esimerkki
esimerkki

Kirjainleikkauksen leventäminen tekee tekstistä samalla lihavampaa, koska kirjainten viivat tulevat vahvemiksi leveys suunnassa. Korkeussuunnassa vahvuus säilyy ennallaan, joten kirjainten viivakонтраст muuttuu. Kirjainleikkauksen kutistaminen vaikuttaa lihavuuteen päinvastaisesti. Tällainen mekaaninen fontin ”korjaileminen” ei välttämättä tuota typografisesti kovin hyvää jälkeä mutta voi sopia lievästi käytettynä esimerkiksi otsikoihin. Tämän oppaan tasalevyistä kirjainperhettä on hieman kutistettu, koska alkuperäinen on kohtuuttoman leveä (esimerkki) verrattuna muiden kirjainperheiden leveyteen.

Jos kirjainperheeseen ei sisälly sopivaa lihavoitua tai kaltevaa leikkausta, voi sellaiset koettaa tehdä myös keinotekoisesti `FakeBold`- ja `FakeSlant`-valitsimien avulla. Niille annetaan arvoksi desimaalilukukerroin. Valitettavasti `FakeBold` toimii vain Xelatex-kääntäjän kanssa.

```
pysty {\addfontfeatures{FakeSlant=.2} kalteva}
```

⇒ pysty *kalteva*

Mekaanisen lihavoinnin tai kallistuksen saa osaksi kirjainperhettä, kun asettaa kirjainperheen määrittelyn yhteydessä halutun kirjainleikkauksen valitsimella `BoldFont` tai `SlantedFont` ja määrittää niille erityisiä ominaisuuksia valitsimella `BoldFeatures` tai `SlantedFeatures`. Seuraava

esimerkki asettaa dokumentin perusfontille kaltevan leikkauksen. Tämän jälkeen komento `\slshape` valitsee *kaltevan* leikkauksen, joka on siis eri asia kuin *kursiivi* (`\itshape`).

```
\setmainfont{...}[
  SlantedFont={...}, % tähän pystyasentoinen leikkaus
  SlantedFeatures={FakeSlant=.2}]
```

3.4.11 Keinotekoinen pienversaali

Moniin fontteihin ei sisälly lainkaan pienversaalia. Jos sellaisen silti haluaa omaan korostusvalikoimaan, voi yrittää pienentää versaalikirjaimet sopivaan kokoon. Seuraavassa esimerkissä versaaleja ensin pienennetään ja sitten venytetään hieman leveysuunnassa. Lisäksi merkikivälejä harvennetaan lievästi. Käytännössä versaaleja pienentämällä ei saada tyylikästä pienversaalia aikaan, koska merkkien viivoista tulee liian ohuita.

```
\textsc{pienversaali} oikea \\
{\addfontfeatures{ScaleAgain=.68, FakeStretch=1.17, LetterSpace=4}
  PIENVERSAALI} keinotekoinen
```

⇒ PIENVERSAALI oikea
PIENVERSAALI keinotekoinen

3.4.12 Matematiikkatilan fontti

Latexin matematiikkatilan (luku 4.16) fontti asetetaan eri tavalla kuin normaalin tilan eli tekstitilan fontit. Matematiikkatilaa varten tarvitaan paketti `unicode-math`¹⁶, joka täytyy ladata muiden matematiikkaan tai fontteihin liittyvien pakettien jälkeen. Paketti tuo komennon `\setmathfont`, joka on matematiikkatilan vastine luvussa 3.4.1 esitellyille fontinmäärittelykomennolle `\setmainfont`, `\setsansfont` ja `\setmonofont`.

```
\usepackage{fontspec}
\usepackage{unicode-math}
\setmainfont{TeX Gyre Pagella} % tekstitilan perusfontti
\setmathfont{TeX Gyre Pagella Math} % matematiikkatilan fontti
```

¹⁶ <https://www.ctan.org/pkg/unicode-math>

Yleistyyli	Lihavointi	Pääteviivaton
<code>math-style=ISO</code>	<code>bold-style=ISO</code>	<code>sans-style=upright</code>
<code>math-style=TeX</code>	<code>bold-style=TeX</code>	<code>sans-style=italic</code>
<code>math-style=french</code>	<code>bold-style=upright</code>	<code>sans-style=literal</code>
<code>math-style=upright</code>		
<code>math-style=literal</code>		

Taulukko 3.10: Matematiikkatilan tyyliasetuksia (`unicode-math`-paketti).

Paketin `unicode-math` lataamisen yhteydessä voi `\usepackage`-komenton valinnaisen argumentin avulla vaikuttaa matematiikkatilassa käytettyihin kirjainleikkauksiin. Argumenttiin sopivia valitsimia ovat `math-style`, `bold-style` ja `sans-style`, ja niille sopivia arvoja on koottu taulukkoon 3.10. Asetukset vaikuttavat esimerkiksi siihen, mitä kirjainleikkausta käytetään, kun ladotaan matematiikkatilan latinalaisia tai kreikkalaisia kirjaimia.

```
\usepackage[math-style=ISO]{unicode-math}
```

Matematiikkatilan fontiksi ei kelpaa mikä tahansa, koska tarvittavien matemaattisten symbolien täytyy sisältyä Open Type -fontin merkki-valikoimaan. Tukea ei ehkä ole, jos sitä ei fontille erikseen mainosteta. Fontitiedoston ominaisuuksia voi tutkia esimerkiksi käyttöjärjestelmän komentotulkissa `otfinfo`-komennolla:

```
otfinfo -s texgyrepagella-math.otf
```

Tex Live -jakelun mukana tulee muutama Open Type -kirjainperhe, joihin kuuluvat myös matematiikan symbolit. Tällaisia kirjainperheitä ovat ainakin Libertinus sekä eri TeX Gyre -perheet. Koko Libertinus-perheen voi ottaa käyttöön hyvin yksinkertaisesti: ladataan vain paketti `libertinus`, joka määrittää kerralla antiikvan, groteskin, tasalevyisen ja matematiikkafontin. Myös tavallisten fontinmäärityskomentojen käyttö on mahdollista.

Esimerkissä 3.9 on komentoja joidenkin matematiikkafonttien lataamiseen. Lähes aina on järkevää käyttää asetusta `Scale=MatchLowercase`, joka asettaa fontin samankokoiseksi kuin vastaava peruskirjainperhe.

```
\setmathfont{TeX Gyre Pagella Math}[Scale=MatchLowercase]
```

```
1 \setmathfont{Libertinus Math Regular}
2 \setmathfont{TeX Gyre Bonum Math}
3 \setmathfont{TeX Gyre DejaVu Math}
4 \setmathfont{TeX Gyre Pagella Math}
5 \setmathfont{TeX Gyre Schola Math}
6 \setmathfont{TeX Gyre Terms Math}
```

Esimerkki 3.9: Matematiikan symboleja sisältäviä vapaita fonteja. Komento `\setmathfont` sisältyy pakettiin `unicode-math`.

3.5 Kieli

Melkein aina Latex-dokumenttiin täytyy ladata kielipaketti ja sen mukana asetukset tiettyjä kieliä varten. Kieliasetukset sisältävät ainakin tavutussäännöt (luku 3.6) sekä kielelle mukautettuja nimiä dokumentin eri osille. Esimerkiksi sisällysluettelon ja kirjallisuusluettelon otsikot tulevat kieliasetuksista, samoin leijuvien taulukoiden ja kuvien nimet ”Taulukko” ja ”Kuva”. Myös muita asetuksia tai komentoja saattaa tulla kieliasetusten mukana, mutta ne vaihtelevat eri kielissä.

Kielipaketteja on kaksi – `babel` ja `polyglossia` –, ja kirjoittajan täytyy valita niistä jompikumpi. Pakettien historia ja kehitys on suunnilleen seuraavanlainen: `babel` on paljon vanhempi, ja monet vanhat oppaat ja esimerkit käsittelevät pelkästään sitä. Kun Latex siirtyi Unicode-aikaan Xelatex-kääntäjän ja `fontspec`-paketin myötä, `babel` ei pysynyt kehityksessä mukana. Syntyi `polyglossia`, joka hallitsi Unicoden sekä muitakin kuin latinalaiseen kirjaimistoon perustuvia kieliä ja kirjoitusjärjestelmiä. `babel`-paketin kehitys ei kuitenkaan pysähtynyt, ja sittemmin myös se on kehittynyt Unicode-aikakaudelle.

Suomen kielen kannalta ei ole merkitystä, kumpaa kielipakettia käyttää, mutta meidän näkökulmastamme ”erikoisemmat” kielet ja kirjoitusjärjestelmät voivat vaatia selvittämistä, kumpi kielipaketti soveltuu paremmin. `babel` sisältää enemmän ominaisuuksia, esimerkiksi omien kommentojen ja muiden viritysten tekemiseen; `polyglossia` on yksinkertaisempi paketti, jonka kehitys tuntuisi keskittyvän vain ydintehtävään.

Seuraavissa alaluvuissa käsitellään kummankin kielipaketin tärkeimmät toiminnot eli kielen valintaan liittyvät asiat. Yleinen ajatus on se,

että Latex-dokumentin esittelyosassa ladataan kielipaketti ja määritellään dokumentin pääasiallinen kieli ja mahdolliset muut kielet. Jos dokumentin tekstiosa sisältää muita kuin pääasiallista kieltä, täytyy käyttää erityisiä komentoja tai ympäristöjä, joilla kerrotaan Latexille, mistä kielestä on kyse.

3.5.1 Polyglossia

Kielipaketti `polyglossia`¹⁷ vaatii toimiakseen Lualatex- tai Xelatex-kääntäjän. Se ei siis toimi perinteisillä Latexin kääntäjillä. Kielipaketin käyttöönotto dokumentin esittelyosassa näyttää esimerkiksi seuraavanlaiselta:

```
\usepackage{polyglossia}
\setdefaultlanguage{finnish}
\setotherlanguage{english}
\setotherlanguage{greek}
```

Eri kielillä on erilaisia valinnaisia asetuksia, jotka täytyy selvittää `polyglossia`-paketin ohjekirjasta. Asetukset liittyvät esimerkiksi kielen kirjoitusjärjestelmän alueelliseen tai historialliseen vaihteluun. Yleinen moniin kieliin sopiva valitsin on `babelshorthands`, joka lisää erilaisia, lähinnä tavutukseen liittyviä komentoja (luku 3.6). Suomen kielen voisi siis ottaa käyttöön myös seuraavalla tavalla:

```
\setdefaultlanguage[babelshorthands]{finnish}
```

Dokumentin tekstiosa käyttää kieltä, joka määriteltiin komennolla `\setdefaultlanguage`. Muita dokumentin esittelyosassa määriteltyjä kieliä voi käyttää väliaikaisesti komennolla, joka alkaa kirjaimilla `\text` ja jatkuu kielen nimellä, esimerkiksi `\textenglish` tai `\textgreek`.

Englannin sana `\textenglish{shorthand}` tarkoittaa 'pikakirjoitusta'.

Toinen vaihtoehto on käyttää kielen nimen mukaista ympäristöä:

```
\begin{greek}
...
\end{greek}
```

¹⁷ <https://www.ctan.org/pkg/polyglossia>

Tietynkieliseksi merkitty teksti voi näyttää ladotussa dokumentissa ihan samalta kuin muukin teksti. Tavutuksessa kuitenkin käytetään eri kielissä eri asetuksia, ja joissakin kielissä voi olla myös pieniä typografisia yksityiskohtia eri tavalla.

Oletuksena kaikilla kielillä käytetään samoja fontteja, mutta kirjoittaja voi määrittellä fontin myös kielikohtaisesti. Tietyn kielen fontti pitää määrittää komentoon, jonka nimessä on ensin kielen nimi ja lopuksi sana font (peruskirjainperhe), fontsf (pääteviivaton) tai fonttt (tasalevyinen). Esimerkiksi fonttikomennot englannin kielelle ja kolmelle eri kirjainperheelle ovat `\englishfont`, `\englishfontsf` ja `\englishfonttt`.

Näiden fonttikomentojen ja kirjainperheiden määrittämiseen kannattaa käyttää `fontspec`-paketin komentoa `\newfontfamily` (luku 3.4.1). Seuraavassa esimerkissä asetetaan kaikki kolme kirjainperhettä kreikan kielelle:

```
\newfontfamily{\greekfont} {GFS Artemisia} [Scale=MatchLowercase]
\newfontfamily{\greekfontsf}{GFS Neohellenic}[Scale=MatchLowercase]
\newfontfamily{\greekfonttt}{TeX Gyre Cursor}[Scale=MatchLowercase]
```

Kirjoittajan ei tarvitse itse käyttää edellä mainittuja komentoja kuten `\englishfont`, `\greekfont` jne. Kirjoittaja käyttää vain kielenvaihtokomentoja, ja fonttikin vaihtuu samalla itsestään, jos vain kohdekielelle ja -kirjainperheelle on määritelty sopiva fonttikomento.

3.5.2 Babel

Vanha kunnon `babel`^{18,19} toimii useiden eri Latex-kääntäjien kanssa. Ennen dokumentit kirjoitettiin rajallisilla merkistöillä (kuten ISO-8859-1), jotka sisältävät vain reilut kaksisataa ihmiskielten kirjoitusmerkkiä. `babel` kuitenkin toimii myös LuaLatex- ja XeLatex-kääntäjillä eli Unicode-merkistön kanssa.

Kieliasetukset otetaan käyttöön `babel`-paketissa seuraavan esimerkin tavoin. Paketin lataamisessa valinnaiseen argumenttiin kirjoitetaan laadattavien kielten nimet ja viimeisenä mainitaan se kieli, joka halutaan pääasialliseksi kieleksi. Toisaalta pääasiallisen kielen voi valita myös

¹⁸ <https://www.ctan.org/pkg/babel>

¹⁹ Katso myös `babel`-paketin mukana toimitettavat kielikohtaiset dokumentit, esimerkiksi tiedosto `finnish.pdf`.

main-valitsimella. Seuraavassa on esimerkki kummastakin kieltenvalintatavasta:

```
\usepackage[english,greek,finnish]{babel}
\usepackage[main=finnish,english,greek]{babel}
```

Dokumentin tekstissä yksittäiset vieraskieliset sanat tai ilmaukset merkitään komennolla `\foreignlanguage`. Komennon ensimmäinen argumentti on kielen nimi ja toinen on sillä kielellä ladottava teksti.

```
Englannin sana \foreignlanguage{english}{shorthand} tarkoittaa
'pikakirjoitusta'.
```

Komennon sijasta voi käyttää ympäristöä `otherlanguage`, joka vaihtaa kieltä ympäristön ajaksi.

```
\begin{otherlanguage}{greek}
...
\end{otherlanguage}
```

Kielen vaihtamista voi helpottaa `\babeltags`-komennolla, joka määrittelee lyhempiä komentoja ja ympäristöjä kielen vaihtamiseen. Komentoa käytetään seuraavan esimerkin tavoin. Sen jälkeen voi käyttää englanninkielisen tekstin merkitsemiseen komentoa `\texteng` tai ympäristöä `eng`.

```
\babeltags{eng = english}
```

Oletuskieli voidaan vaihtaa kesken dokumentin komennolla `\selectlanguage`. Komennon argumentiksi annetaan kielen nimi, esimerkiksi seuraavalla tavalla:

```
\selectlanguage{english}
```

Eri kielille on mahdollista asettaa eri kirjainperheet `\babelfont`-komennolla. Tämä komento korvaa `fontspec`-paketin fontinvalintakomennot (luku 3.4.1) ja muodostaa uuden korkeamman tason komennon, joka sisältää myös kielen. Jos siis käyttää komentoa `\babelfont`, ei saa samanaikaisesti käyttää komentoja `\setmainfont`, `\setsansfont` eikä `\setmonofont`.

```

1 \belfont{rm}{TeX Gyre Termes}
2 \belfont{sf}[Scale=MatchLowercase]{TeX Gyre Heros}
3 \belfont{tt}[Scale=MatchLowercase]{TeX Gyre Cursor}
4
5 \belfont[greek]{rm}[Scale=MatchLowercase]{GFS Artemisia}
6 \belfont[greek]{sf}[Scale=MatchLowercase]{GFS Neohellenic}

```

Esimerkki 3.10: `\belfont`-komennon käyttö dokumentin kirjainperheiden valintaan ja kielikohtaisten kirjainperheiden valintaan.

Oman dokumentin kirjainperheet voitaisiin määrittää `\belfont`-komennolla esimerkin 3.10 tavoin. Ensin esimerkissä määritellään peruskirjainperhe (`rm`), pääteviivaton perhe (`sf`) ja tasalevyinen perhe (`tt`). Sen jälkeen määritellään kreikan kielessä (`greek`) käytettävä peruskirjainperhe ja pääteviivaton perhe.

Esimerkissä ennen fontin nimeä (esim. TeX Gyre Heros) oleva valinnainen argumentti on sama kuin `fontspec`-paketin fontinvalintakomennossa. Sen avulla määritellään kyseisen kirjainperheen asetuksia. Lisätietoa voi lukea fontteja käsittelevästä luvusta 3.4 ja `fontspec`²⁰-paketin ohjekirjasta.

3.6 Tavutus

Tex tavuttaa eli katkaisee sanoja automaattisesti rivien lopussa, jotta se saa tekstikappaleet näyttämään tasapainoisilta. Lähtökohtaisesti tavutus määräytyy kielikohtaisten tavutussääntöjen ja -asetusten perusteella, mutta kirjoittaja voi tehdä poikkeuksia kirjoittamalla tavutusvihjeitä. Käytännössä tavutusvihjeitä tarvitaan välillä. Kielen valintaa ja yleisiä kieliasetuksia käsitellään tarkemmin luvussa 3.5.

3.6.1 Yleiset tavutussäännöt

Texin automaattinen tavutus ei perustu varsinaiseen sanojen eikä taivutusmuotojen tunnistamiseen vaan yksinkertaisiin kirjainpohjaisiin

²⁰ <https://www.ctan.org/pkg/fontspec>

sääntöihin. Säännöt pyrkivät kuvaamaan kielen tavujen rakenteen ja huomioimaan myös typografiaan liittyviä suosituksia.

Automaattinen tavutus auttaa paljon, mutta se ei yksinään riitä, koska se tavuttaa välillä kielen kannalta väärin tai tuottaa muuten suositusten vastaista jälkeä. Kirjoittajan täytyy siis auttaa välillä eli kirjoittaa tavutusvihjeitä. Yksi tapa tavutusvihjeiden kirjoittamiseen on `\hyphenation`-komento, jolla määritellään yksittäisten sanojen tavutuskohdat kaikkialla dokumentissa. Seuraava esimerkki selventää komennon käyttöä:

```
\hyphenation{
  ala-indek-si alku-osa
  nimen-omaan
  typo-gra-fi-nen
}
```

Komennon perään aaltosulkeiden sisään kirjoitetaan sanoja, jotka erotetaan toisistaan sanaväleillä. Sanoihin kirjoitetaan yhdysmerkki niihin kohtiin, joista sanan katkaiseminen on sallittua. Jos sanassa itsessään on yhdysmerkki, sen tavutusta ei voi käsitellä tällä komennolla. Luvussa 3.6.2 kerrotaan muita tapoja.

`\hyphenation`-komennon voi sijoittaa dokumentin esittelyosaan tai tekstiosaan, mutta sijainti vaikuttaa sen toimintaan. Jos komennon sijoittaa dokumentin esittelyosaan ennen kuin mitään kieltä on ladattu tai valittu, se vaikuttaa kaikkien sanojen tavutukseen kielestä riippumatta. Jos komennon sijoittaa dokumentin tekstiosaan eli kielen valitsemisen jälkeen, se vaikuttaa vain kyseisen kielen eli yleensä dokumentin pääasiallisen kielen tavutukseen.

3.6.2 Yksittäisten sanojen tavutus

Tietyt sanassa mukana olevat merkit kytkevät tavalliset tavutussäännöt pois päältä. Jos sanassa on mukana yksikin yhdysmerkki (-), lyhyt ajatusviiva (--) tai pitkä ajatusviiva (---), sana katkaistaan vain näiden merkkien jälkeen, eli muut tavutuskohdat kytetään pois käytöstä.

Myös tavutusvihje (\-) käyttäytyy yleensä samoin: jos sanassa on mukana yksikin tavutusvihje, tavalliset tavutuskohdat kytkeytyvät pois käytöstä, ja sana katkaistaan vain tavutusvihjeiden kohdalta sekä edel-

Lähde	Tavutus	Merkitys
matkustaa	matkustaa	tavutus kaikista kohdista
matkus\ -taa	matkustaa	vain tavutusvihjeen kohdalta
matkus\ -taa	matkustaa	(babel ja suomen kieli)
matka-aika	matka-aika	vain yhdysmerkin jälkeen
matka-ai\ -ka	matka-aika	vain yhdysmerkki ja tavutusvihje
Oulu--Rovaniemi	Oulu-Rovaniemi	vain ajatusviivan jälkeen
Oulu--Rova\ -niemi	Oulu-Rovaniemi	vain ajatusviiva ja tavutusvihje
Oulu--Rova\ -niemi	Oulu-Rovaniemi	(babel ja suomen kieli)
matka-	matka-	ei tavutuskohtia
-aika	-aika	vain yhdysmerkin jälkeen

Taulukko 3.11: Tavutusvihjeen, yhdysmerkin ja ajatusviivan vaikutus tavutukseen. `babel`-kielipaketin tavutusvihje (`\-`) ei suomen kielessä estä sanan tavutusta muista kohdista.

lä mainittujen viivaväli-merkkien jälkeen. Kielipaketti `babel` käyttäytyy kuitenkin suomen kielessä eri tavoin. Se määrittelee suomen kielelle tavutusvihjeen siten, että se sallii sanan tavutuksen muistakin kohdista kuin tavutusvihjeen kohdalta.

Taulukossa 3.11 on esimerkkejä tavutusvihjeiden, yhdysmerkin ja ajatusviivan vaikutuksesta. Ensimmäisessä sarakkeessa on esimerkkinä siinä muodossa kuin se kirjoitetaan lähdetiedostoon. Toisessa sarakkeessa on ladottu sana, johon on pystyviivalla merkittynä mahdolliset tavutuskohdat.

Taulukon viimeinen rivi paljastaa suomen kielen kannalta ongelmallisen tilanteen. Esimerkiksi ilmauksessa *matkasuunnitelma ja -aika* ei riviä saa katkaista sanassa *-aika* olevan yhdysmerkin jälkeen, koska rivin loppuun jäisi yksinäinen yhdysmerkki. Latexin peruskeinoilla sana täytyy laittaa näkymättömään laatikkoon, joka pitää merkit yhdessä: `\mbox{-aika}`. Toinen vaihtoehto on sitovan yhdysmerkin käyttö, jota käsitellään seuraavassa alaluvussa.

3.6.3 Tavutuksen sallivia ja sitovia merkkejä

Unicode-merkistön sitova yhdysmerkki `U+2011 NON-BREAKING HYPHEN` näkyy tavallisena yhdysmerkkinä, mutta se estää sanan katkaisemisen yhdysmerkin vierestä. `U+2011`-merkkiä voi käyttää, mutta se

Komento	Merkitys
<code>\-/</code>	tavutuksen salliva yhdysmerkki
<code>\=/</code>	sitova, tavutuksen salliva yhdysmerkki
<code>\--</code>	tavutuksen salliva lyhyt ajatusviiva
<code>\==</code>	sitova, tavutuksen salliva lyhyt ajatusviiva
<code>\---</code>	tavutuksen salliva pitkä ajatusviiva
<code>\===</code>	sitova, tavutuksen salliva pitkä ajatusviiva

Taulukko 3.12: `extdash`-paketin komentoja.

ei ole aina toiminut luotettavasti eikä yhdenmukaisesti eri kääntäjissä. Yhteensopivuussyistä voi olla järkevää käyttää muita vaihtoehtoja sitovalle yhdysmerkille.

Paketti `extdash`²¹ tuo uusia komentoja ja mahdollisuuksia tavutuksen hallintaan. Komennot ovat sellaisia kuin `\Hyphdash` ja `\Endash`, mutta niille on saatavilla myös lyhemmät muodot, jos paketin lataa käyttämällä `shortcuts`-valitsinta.

```
\usepackage[shortcuts]{extdash}
```

Paketti sisältää kaksi lisävaihtoehtoa kolmelle viivävälimerkille eli yhdysmerkille, lyhyelle ajatusviivalle ja pitkälle ajatusviivalle. Kun Texin viivävälimerkit (luku 3.6.2) aina estävät tavutuksen muualta kuin välimerkin jälkeen, `extdash`-paketin perusvaihtoehdot sallivat tavutuksen muualtakin. Lisäksi kaikille kolmelle viivävälimerkille on sitova versio, joka estää tavutuksen välimerkin kohdalta (mutta sallii muualta).

Taulukossa 3.12 ovat `extdash`-paketin tärkeimmät komennot ja niiden merkitykset. Taulukossa 3.13 vertaillaan `extdash`-paketin komentoja Texin vastaaviin.

On tärkeää huomioida, että `extdash`-paketin komennot ovat todellakin normaaleja komentoja. Se tarkoittaa, että komennon jälkeiset sanavälit syödään pois. Tämä asia saattaa unohtua seuraavanlaisessa tilanteessa, jossa käytetään tavutuksen sallivaa yhdysmerkkikomentoa:

```
matka\-/ ja aika\-/arvio
```

⇒ matka-ja aika-arvio

²¹ <https://www.ctan.org/pkg/extdash>

Lähde	Tavutus	Lähde	Tavutus
matka-aika	matka- aika	Oulu--Rovaniemi	Oulu- Rovaniemi
matka\-/aika	matka- aika	Oulu\--Rovaniemi	Oulu- Rovaniemi
matka\=/aika	matka- aika	Oulu\==Rovaniemi	Oulu- Rovaniemi
matka-	matka-	-aika	-aika
matka\-/	matka-	\=/aika	-aika

Taulukko 3.13: Texin ja `extdash`-paketin komentojen vertailua.

Ensimmäisen yhdysmerkkikomennon jälkeinen sanaväli hävisi, ja syntyi virheellinen sana *matka-ja*. Sanavälin saa säilymään, kun kirjoittaa komennon perään aaltosuljeparin tai kenoviivan ja sanavälin.

```
matka\-\{} ja aika\-/arvio \\  
matka\-\ / ja aika\-/arvio
```

⇒ matka- ja aika-arvio
matka- ja aika-arvio

3.6.4 Tavutus sanan reunasta

Asetukset `\lefthyphenmin N` ja `\righthyphenmin N` vaikuttavat tavutukseen sanan reunoissa. Argumentti `N` on positiivinen kokonaisluku, ja se määrittelee, kuinka monta merkkiä vähintään sanan vasemmasta tai oikeasta reunasta pidetään yhdessä. Oletusarvot ovat kielikohtaisia ja määritellään `polyglossia`- ja `babel`-paketeissa. Suomen kielessä kumpikin asetus on kaksi (2) merkkiä.

Nämä asetukset alustetaan kielikohtaisiin oletusarvoihin aina, kun kieliasetukset tulevat voimaan. Näin on esimerkiksi dokumentin aloitettavan `document`-ympäristön alussa ja aina kielen vaihtuessa. Jos asetuksia haluaa muuttaa, täytyy omat muutokset tehdä aina edellä mainittujen asioiden jälkeen. Toisaalta vielä kätevempää on sisällyttää omat asetukset kielikohtaisiin alustuskomentoihin. Sillä tavoin omat asetukset tulevat voimaan automaattisesti kielen vaihtuessa.

```
\addto{\captionsfinnish}{  
  \lefthyphenmin 3  
  \righthyphenmin 3  
}
```

Edellisessä esimerkissä oleva komento `\addto` on `polyglossia`- ja `babel`-paketin ominaisuus, jolla lisätään omia komentoja kielikohtaisiin asetuksiin. Tässä esimerkissä käsitellään suomen kielen asetuksia (`\captionsoffinnish`). Komento täytyy suorittaa lähdedokumentin esittely-osassa.

3.6.5 Muita tavutusasetuksia ja -vinkkejä

Aiemmin mainittu `extdash`-paketti sisältää tavutuksen hallintaan liittyviä välimerkkejä, mutta kielestä riippuen sellaisia on myös kielipaketeissa `babel` ja `polyglossia`. Jälkimmäiseen pakettiin lisävälimerkit täytyy kytkeä erikseen päälle käyttämällä valitsinta `babelshorthands` (luku 3.5.1).

Molemmat kielipaketit tuovat suomen kieleen muun muassa sitovan, tavutuksen sallivan yhdysmerkin "-", joka toimii samoin kuin `extdash`-paketin komento `\=/`. Lisäksi on tavutusvihje "-", joka ei tee yhdysmerkkiä katkaisukohtaan. Sitä voi käyttää teknisten ilmausten tavutusvihjeenä, koska niihin ei välttämättä haluta yhdysmerkkiä edes katkaisutilanteessa. Muitakin yleislainausmerkillä (") alkavia erikoismerkkejä sisältyy kieliasetuksiin.

Varsinainen tavutuksen peruskomento on `\discretionary`, joka mahdollistaa omanlaisten tavutuskohtien määrittelyn. Komento kirjoitetaan sanassa juuri siihen kohtaan, johon tavutuskohta halutaan, ja komennon argumenttien rakenne on seuraavanlainen:

```
\discretionary{loppu}{alku}{katkaisematon}
```

Komennon ensimmäinen argumentti `loppu` ilmaisee katkaisutilanteessa rivin loppuun ladottavat merkit. Toinen argumentti `alku` ilmaisee katkaisutilanteessa seuraavan rivin alkuun ladottavat merkit, ja kolmas argumentti tarkoittaa katkaisemattomaan sanaan ladottavia merkkejä. Normaali tavutuskohta sanaan *tavu* määriteltäisiin seuraavan esimerkin tavoin:

```
ta\discretionary{-}{}{}vu
```

Edellisessä esimerkissä komennon ensimmäinen argumentti on yhdysmerkki, koska rivin loppuun tietenkin halutaan yhdysmerkki silloin,

kun sana katkaistaan tästä kohdasta. Toinen argumentti on tyhjä, koska seuraavan rivin alkuun ei kirjoiteta suomen kielessä mitään ylimääräistä. Myös kolmas argumentti on tyhjä, koska katkaisemattomaan sanaan ei haluta mitään merkkiä tavujen väliin.

`\discretionary`-komennolla voi luoda myös tavutuskohtia, joihin ei ilmesty yhdysmerkkiä eikä mitään muutakaan merkkiä katkaisutilanteessa. Tämä vastaa kielipakettien tavutusvihjettä `"`. `\discretionary`-komennolla sama toteutetaan jättämällä kaikki argumentit tyhjäksi:

```
\discretionary{}{}{}
```

Joissakin kielissä sanan katkaiseminen yhdysmerkin kohdalta vaatii, että rivin loppuun kirjoitetaan yksi yhdysmerkki ja seuraavan rivin alkuun toinen. Näin ilmaistaan, että sanassa on pysyvä yhdysmerkki eikä vain väliaikaisesti katkaisun merkinä. Tällaisia kieliä varten saattaa kielipaketissa olla omat yhdysmerkkitoiminnot tai oikeanlainen tavutuslogiikka voi jo sisältyä kieliasetuksiin, mutta tällaisen tavutuskohdan saa myös seuraavalla tavalla:

```
\discretionary{-}{-}{-}
```

Tavutuksen suunnittelussa ja tutkimisessa voi auttaa `showhyphens`²²-paketti. Kun paketti on ladattuna, kaikkien sanojen mahdollisiin tavutuskohtiin ladotaan ohut punainen pystyviiva. Paketti hyödyntää LuaLatex-kääntäjän ominaisuuksia, eikä se siis toimi muiden kääntäjien kanssa.

3.6.6 Suomen kielen tavutus

Texin kirjainyhdistelmiin perustuvat tavutussäännöt eivät yksinään ole tarpeeksi älykkäitä suomen kieleen, ja esimerkiksi yhdyssanojen rajakohdat tuottavat usein ongelmia. Sana *alkuosa* katkaistaan Latexissa kohdista *al-kuo-sa*. Se on kyllä oikein tavurakenteen kannalta mutta käytännössä ongelmallinen. Tässä ei ole kyse *uo*-diftongista eli samaan tavuun kuuluvista vokaaleista, vaan yhdyssanan rajalla on myös tavutuskohta (*al-ku-o-sa*). Lisäksi sanaa ei saisi katkaista siten, että siitä jää yksittäinen kirjain eri riville (ei: *o-sa*).

²² <https://www.ctan.org/pkg/showhyphens>

Parasta olisi katkaista suomen kielen yhdyssanat vain yhdysosien välistä (*alku-osa*). Muuallakin voi katkaista (*al-ku-osa*), kunhan sanasta eikä sen yhdysosasta ei jää yksittäinen kirjain eri riville. Mielellään ei katkaista myöskään kahden vokaalin välistä, jos ne kuuluvat samaan sanaan (ei: *kau-emmin*). Joskus halutaan välttää myös niin sanottuja orpotavuja eli sitä, että tekstikappaleen viimeiselle riville jää vain yksi tavu.

Käytännössä siis suomenkielinen teksti ja hyvä typografia vaativat välillä tavutusvihjeiden kirjoittamista. Yhdyssanojen osien väliin tarvitaan tavutusvihje silloin, kun jälkimmäinen osa alkaa vokaalilla tai useammalla kuin yhdellä konsonantilla. Joitakin tällaisia tapauksia Tex osaa tavuttaa oikein ilman tavutusvihjeitäkin, mutta etukäteen sitä ei tiedä, ellei ole kokemusta.

```
alku\-osa pusku\-traktori
```

Yhdysmerkin tai ajatusviivan sisältävät pitkät yhdyssanat voivat vaatia tavutuskohtien lisäämistä, koska yhdysmerkki ja ajatusviivat estävät tavutuksen muualta kuin näiden merkkien jälkeen. Ilman tavutuskohtien lisäämistä Texillä ei ehkä ole riittävästi vaihtoehtoja tekstikappaleen rivittämiseen. Voi syntyä liian suuria sanavälejä, tai joistakin riveistä tulee ylipitkiä, eli ne yltävät marginaalin puolelle.

Tavutuskohtia voi lisätä sopiviin kohtiin tavutusvihjeillä. Vaihtoehtoisesti kaikki yleisten tavutussääntöjen mukaiset tavutuskohdat saa käyttöön `extdash`-paketin välimerkkikomennoilla (taulukko 3.12). Seuraavassa esimerkissä on tavutuksen hallintaa sanalle *Molotov–Ribbentrop-sopimus*:

```
Molo\-tov--Ribben\-trop-sopi\-mus % tavutusvihjeet
Molotov\--Ribben\-trop\-/sopimus % tavutuksen sallivat välimerkit
```

`\discretionary`-komentoa (luku 3.6.5) voi hyödyntää suomen kielen sanoissa silloin, kun niissä on heittomerkki erottamassa kahta tavurajan molemmin puolin olevaa samaa vokaalia, esimerkiksi sanoissa *vaa'an* ja *liu'uttaa*. Normaalisti Tex ei katkaise heittomerkin kohdalta lainkaan, eikä se olisi suomen kielessä suositeltavaakaan, koska vokaalien välistä ei mielellään katkaista sanaa. Jos tavutuksen kuitenkin haluaa myös

heittomerkin kohdalle, täytyy huolehtia, että tavutustilanteessa heittomerkki poistuu ja sen paikalle tulee yhdysmerkki rivin loppuun. Sanan *vaa'an* voisi siis kirjoittaa seuraavalla tavalla:

```
vaa\discretionary{-}{}{'}an
```

Komennon kolmas argumentti on heittomerkki, koska se pitää latoa tavurajalle silloin, kun sanaa ei katkaista tästä kohdasta. Mikäli tällaisia tarvitsee paljon, on järkevää määritellä sitä varten yksinkertaisempi komento, jota sitten käytetään sanoissa heittomerkin sijasta.

```
\newcommand{\hm}{\discretionary{-}{}{'}  
vaa\hm an
```

Suomen kielessä käytetään heittomerkkiä myös taivutuspäätteen, liitteen tai johtimen edellä silloin, kun sanavartalon kirjoitusasu päättyy konsonanttiin mutta ääntöasu vokaaliin, esimerkiksi *show'ssa*. Näissä tilanteissa ei ole kyse tavurajasta vaan morfeemirajasta eli merkityksellisten osien rajakohdasta. Tavurajakin voi sattua samaan paikkaan, mutta tavutettaessa heittomerkki säilyy: *show'-hun*.²³ Mikäli tällainen tavutuskohta halutaan mukaan, käytetään sanassa tavallista tavutusvihjettä: *show'\-hun*. Mieluummin ei kuitenkaan katkaista sanoja heittomerkin kohdalta.

3.6.7 Tavutus pois päältä

Tavutuksen voi kytkeä kokonaan pois päältä **polyglossia**-kielipaketin komennolla `\disablehyphenation`. Tavutuksen saa takaisin päälle komennolla `\enablehyphenation`.

Toinen vaihtoehto on käyttää fontin asetusta `HyphenChar=None` kirjainperheen määrittelyn yhteydessä. Oletuksena tavutus on pois päältä tasalevyisestä fontista, eli esimerkiksi `\texttt`-komennon argumenttina olevaa tekstiä ei tavuteta. Fontteja ja niiden asetuksia käsitellään luvussa 3.4.

Käytännössä tavutus menee pois päältä myös silloin, kun tekstikapaleet tasaa vain vasempaan reunaan eli tekee oikealle liehureunan ko-

²³ Asiaa ei yleensä mainita kielenhuolto-oppaissa. Tieto perustuu Kielikello-lehden 2/2006 artikkeliin: <https://www.kielikello.fi/-/lainausmerkit->. Viittäuspäivä 6.7.2020.

mennolla `\raggedright`. Tekstikappaleiden tasaamista ja palstan muotoa käsitellään luvussa 4.1.1.

Luku 4

Rakenne ja sisältö

Tämä luku on oppaan kaikista luvuista kenties käytännönläheisin. Luku keskittyy asioihin, joita kirjoittaja miettii erityisesti dokumentin sisällön kirjoitusvaiheessa. Kirjoittaja tekee esimerkiksi valintoja otsikoinnin ja muun jäsentämisen osalta. Hän miettii tiedon esittämistä paitsi tekstin avulla myös esimerkiksi luetelmien, kuvien ja taulukoiden avulla. Kirjoittaja tekee myös typografisia valintoja tekstin muotoilun ja korostuskeinojen näkökulmasta. Edellä mainittuja ja muitakin dokumentin rakenteen ja sisällön asioita käsitellään niin tekniikan kuin typografiankin näkökulmasta.

4.1 Tekstikappaleet

Tekstikappale on tekstin osa, jonka pitäisi käsitellä suunnilleen yhtä asiakokonaisuutta. Se voi olla esimerkiksi yksi aihe, näkökulma, ajan kohta tai henkilö. Tekstin seuraava kappale käsittelee jotakin toista aihetta, näkökulmaa tms. Kappaleen vaihtuminen on lukijalle merkki siitä, että tekstin sisällössäkään jokin muuttuu.

Latexin lähdetiedostoissa kappaleen vaihtuminen ilmaistaan kirjoittamalla kappaleiden väliin vähintään yksi tyhjä rivi. Tätä merkintäkielellä käsitellään myös luvussa 2.1.4. Kappale vaihtuu myös komentamalla `\par`, joka sopii käytettäväksi esimerkiksi kommentojen määrittelyssä (luku 2.2), kun halutaan varmistaa kappaleen vaihtuminen tietyssä kohdassa.

Ladotuissa teksteissä kuten kirjoissa ja lehdissä kappaleen vaihtumisen ilmaistaan melkein aina siten, että uuden kappaleen ensimmäinen rivi sisennetään hieman. Niin on tässäkin oppaassa. Toisinaan tekstikappaleet erotetaan pystysuuntaisella välillä, ja silloin kappaleiden ensimmäistä riviä ei sisennetä. Kappaleiden välejä, sisennyksiä, rivien taasaamista ja muita asetuksia käsitellään seuraavissa alaluvuissa.

Monissa kappaleisiin liittyvissä asetuksissa tarvitaan Texin mittoja ja mittayksiköitä. Mittoihin liittyvää tekniikkaa käsitellään tarkemmin luvussa 2.4, joka on syytä tuntea ennen tämän alaluvun lukemista.

4.1.1 Tasaaminen ja palstan muoto

Perusdokumenttiluokissa (luku 3.1.1) tekstikappaleet tasataan oletuksena palstan molempiin reunoihin, ja tätä palstan muotoa kutsutaan tasapalstaksi. Se tarkoittaa samalla sitä, että rivillä olevia sanavälejä venytetään sopivasti, jotta jokainen rivi näyttäisi yhtä pitkältä ja palstan molemmat reunat tasaiselta.

Käytännössä sanavälien venymiselle on määritelty yläraja, jonka yli niitä ei venytetä. Ylärajan tarkoituksena on estää liian suuret ja rumat sanavälit. Rajoitus on sinänsä järkevä, mutta se voi myös johtaa siihen, että Tex ei saa tasattua kaikkia tekstikappaleita palstan oikeasta reunasta: jotkin rivit yltyvät palstan reunan yli; jotkin rivit jäävät vajaaksi. Näin käy usein varsinkin suomen kielessä, jonka sanat ovat usein pitkiä ja riveillä on vähänlaisesti sanavälejä. Suomen kielessä sanavälien venymisen yläraja on usein tarpeellista asettaa oletusarvoa suuremmaksi. Se tehdään mitan `\emergencystretch` avulla, esimerkiksi seuraavasti:

```
\setlength{\emergencystretch}{1em}
```

Kaikenlaiset kappaleiden latomiseen liittyvät tekniset rajoitukset voi poistaa tai asettaa hyvin suuriksi komennolla `\sloppy`. Komento asettaa muun muassa sanavälien venymisen ylärajaksi 3 em. Tämän komennon käyttö ei ole kovin suositeltavaa, koska sillä on muitakin seurauksia ja se voi vaikuttaa myös sellaisiin kappaleisiin, jotka muuten saataisiin ladottua nätisti. Parempi on asettaa vain mitta `\emergencystretch` riittävän suureksi. Sanavälien venymiseen ja kappaleiden tasaiseen latomiseen liittyvät asetukset voi palauttaa oletusarvoihin komennolla `\fussy`.

Komento	Ympäristö	Merkitys
<code>\raggedright</code>	<code>flushleft</code>	vasen tasaus, oikea liehu
<code>\raggedleft</code>	<code>flushright</code>	oikea tasaus, vasen liehu
<code>\centering</code>	<code>center</code>	keskitetty
<code>\RaggedRight</code>	<code>FlushLeft</code>	vasen tasaus, oikea liehu, tavutus (<code>raggedze</code>)
<code>\RaggedLeft</code>	<code>FlushRight</code>	oikea tasaus, vasen liehu, tavutus (<code>raggedze</code>)
<code>\Centering</code>	<code>Center</code>	keskitetty, tavutus (<code>raggedze</code>)
<code>\justifying</code>	<code>justify</code>	tasapalsta, tavutus (<code>raggedze</code>)

Taulukko 4.1: Tekstikappaleen tasaamiseen ja palstan muotoon vaikuttavat komennot ja ympäristöt. Osa sisältyy `raggedze`-pakettiin.

Hyvin tavallista on tasata teksti pelkästään vasempaan reunaan, jolloin rivien pituudet vaihtelevat ja oikealla on niin sanottu liehureuna. Oikea liehureuna sopii pitkiin teksteihin yhtä hyvin kuin tasapalstakin, mutta se on parempi valinta erityisesti silloin, kun palsta on kapea. Nimittäin kapealla palstalla venyviä sanavälejä on käytettävissä hyvin vähän ja oikean reunan tasaaminen vaatii sanavälien venyttämistä joskus kohtuuttoman paljon. Tekstiin jää rumia aukkoja.

Kappaleiden tasaamiseen ja palstan muotoon vaikuttavia komentoja ja ympäristöjä on koottu taulukkoon 4.1. Taulukossa on mainittu ensin Latexin omat komennot ja sitten `raggedze`¹-paketin vastaavat. Latexin omat komennot estävät sanojen tavuttamisen, kun taas `raggedze`-paketin komennot sallivat tavutuksen normaalisti.

4.1.2 Pystysuuntaiset välit

Kappaleiden väliin ladottava pystysuuntainen tyhjä tila asetetaan mitan `\parskip` avulla. Se on oletuksena nolla, mutta pientä venymistä kuitenkin sallitaan, eli joissakin tilanteissa kappaleiden väliin voidaan laotoa pieni tyhjä tila. Jos tyhjää tilaa ei haluta missään tilanteessa, asetetaan mitta vain nolllaksi:

```
\setlength{\parskip}{0ex}
```

Seuraava esimerkkikomento asettaa kappaleväliksi 1,3 ex. Lisäksi se sallii kappalevälin venyä 0,2 ex:n verran tai kutistua 0,1 ex:n verran.

¹ <https://www.ctan.org/pkg/ragged2e>

```
\setlength{\parskip}{1.3ex plus .2ex minus .1ex}
```

Silloin kun kappaleet ladotaan erilleen toisistaan, on yleensä hyvä sallia kappalevälin venyä tai kutistua hieman, koska venyvät pystysuuntaiset välit antavat Texille paremmat mahdollisuudet latoa hyvännäköisiä sivuja. Venyvien välien avulla esimerkiksi sivujen tekstialueen ylä- ja alareunat saadaan aina samalle kohdalle. Toisaalta myös liian suuret ja toisistaan liiaksi poikkeavat kappalevälit voivat olla rumannäköisiä.

Tavallista kappaleväliä suurempien pystysuuntaisten välien tekemiseen on olemassa kolme valmista komentoa: suurimmasta pienimpään ne ovat `\bigskip`, `\medskip` ja `\smallskip`. Ne sopivat käytettäväksi yksittäisiin tilanteisiin, joissa normaali kappaleväli on liian vähän. Jos sivunvaihto osuu näiden komentojen kohdalle, mitään väliä ei ladota sivun loppuun eikä seuraavan alkuun.

Edellä mainittujen komentojen latoman välin suuruuteen voi vaikuttaa mittojen `\bigskipamount`, `\medskipamount` ja `\smallskipamount` avulla. Seuraavassa on esimerkkikomennot mittojen määrittelyyn ja samalla niiden oletusarvot:

```
\setlength{\bigskipamount} {12pt plus 4pt minus 4pt}  
\setlength{\medskipamount} {6pt plus 2pt minus 2pt}  
\setlength{\smallskipamount}{3pt plus 1pt minus 1pt}
```

Komentojen `\bigskip`, `\medskip` ja `\smallskip` sijasta voi käyttää myös komentoja `\bigbreak`, `\medbreak` tai `\smallbreak`. Nämä toimivat lähes samalla tavalla, mutta niihin sisältyy myös sivunvaihtovihje. Toisin sanoen ne vaikuttavat ladonta-algoritmiin siten, että komennon kohdalla todennäköisyys sivun vaihtumiselle kasvaa suhteessa muihin kohtiin. Sivun voi edelleen vaihtua muustakin kohdasta, jos algoritmi löytää omasta mielestään vielä paremman paikan.

Pystysuuntaisten välien yleiskomento on `\vspace`, jolle annetaan argumentiksi välin suuruus ja mahdolliset venymisen rajat. Tämäkin komento jättää välin latomatta, jos se sattuu sivunvaihdon kohdalle. Sen sijaan tähdellinen versio `\vspace*` latoa välin joka tapauksessa, vaikka se olisi sivun lopussa tai alussa.

Tekstikappale.
`\vspace{5ex plus 1ex minus .5ex}`

Toinen tekstikappale.

Komento `\addvspace` toimii lähes samoin kuin `\vspace`, mutta se huomioi mahdolliset peräkkäiset `\addvspace`-komennot ja varmistaa, että vain suurin väli toteutuu. Jos siis useita `\addvspace`-komentoja sattuu peräkkäin, niiden määrittämiä välejä ei ladota peräkkäin vaan ainoastaan suurin niistä ladotaan. Seuraava esimerkki latao kappaleiden väliin 2 ex:n suuruisen pystysuuntaisen välin:

Tekstikappale.

`\addvspace{1ex} \addvspace{2ex} \addvspace{.5ex}`
Toinen tekstikappale.

Jos edellisessä esimerkissä olisi käytetty `\vspace`-komentoa, pystysuuntaisen välin suuruus olisi 3,5 ex, joka on välien yhteenlaskettu suuruus.

`\addvspace`-komento soveltuu hyvin komentojen ja ympäristöjen määrittelyyn (luvut 2.2 ja 2.3). Esimerkiksi itse määritellyn ympäristön alussa ja lopussa voi `\addvspace`-komennolla varmistaa tietynsuuruisen välin, mutta jos sama tai muu vastaava ympäristö on dokumentissa kahdesti peräkkäin, huomioidaan pystysuuntaisen väli vain kerran eli suurimman välin mukaan. Jotkin Latexin valmiit ympäristöt tekevät juuri näin eli käyttävät `\addvspace`-komentoa välien asettamiseen.

Sivun alueella äärettömästi venyvän pystysuuntaisen välin saa komennolla `\vfill`. Mitan luonnollinen arvo on nolla, mutta se voi venyä niin, että se täyttää kaiken tyhjän tilan sivulla. `\vfill`-komento tarkoittaa käytännössä samaa kuin `\vspace{0mm plus 1fill}` -komento. Texin venyviä mittoja ja välejä käsitellään tarkemmin luvussa 2.4.3.

4.1.3 Ensimmäisen rivin sisennys

Kirjojen ja lehtien typografiassa kappaleen ensimmäinen rivi on tapana sisentää merkiksi siitä, että alkaa uusi kappale. Kappaleiden välissä ei ole pystysuuntaista tilaa, vaan pelkkä sisennys on lukijalle merkki siitä, että tekstin sisällössä siirrytään seuraavaan asiaan. Oletusasetuksilla Latex latao sisennyksen automaattisesti kappaleiden alkuun.

Ensimmäisen rivin sisennyksen suuruus asetetaan mitan `\parindent` avulla, seuraavan esimerkin mukaisesti. Sopiva mittayksikkö tähän tarkoitukseen on em, koska se viittaa suoraan nykyisen fontin kokoon.

```
\setlength{\parindent}{1em}
```

Edellä mainittu mitta pitäisi asettaa nolnaan silloin, kun kappaleet erotetaan pystysuuntaisen välin avulla. Välihän jo sinänsä ilmaisee, että kappale vaihtuu, joten sisennys on turha.

```
\setlength{\parskip}{1.3ex plus .2ex minus .1ex}
\setlength{\parindent}{0em} % Ei sisennystä.
```

`\parindent`-mitan levyisen välin voi tehdä mihin tahansa komennolla `\indent`. Tätä komentoa ei tavallisesti tarvita, koska kappaleet alkavat automaattisesti sen suuruisella sisennyksellä. Tarpeellisempi komento on sen vastakohta `\noindent`, joka voidaan kirjoittaa kappaleen alkuun estämään kyseisen kappaleen alun sisentäminen.

```
\noindent
Tämän tekstikappaleen ensimmäistä riviä ei sisennetä.
```

Suomenkielisissä julkaisuissa on tavallista, että leipätekstin kappaleessa ei ole sisennystä, jos sitä ennen on pystysuuntainen väli. Tällainen tilanne on aina otsikoiden jälkeen mutta myös kokonaan sisennetyn tekstikappaleen jälkeen (esim. lohkolainaus, luku 4.1.5) tai kuvan, taulukon, luettelman tai muun vastaavan osan jälkeen, jos nämä ovat osa tekstivirtaa eivätkä leijuvia osia (luku 4.9). Käytäntöön on joskus poikkeuksia suomenkielisessäkin typografiassa, mutta eri kielten välillä käytäntö voi vaihdella enemmänkin.

Latex estää sisennyksen automaattisesti otsikoiden jälkeen mutta latoi sisennyksen kuitenkin kaikkien muiden elementtien ja pystysuuntaisen välin jälkeen. Jos sisennys halutaan estää, pitäisi kyseiset kappaleet aloittaa aina `\noindent`-komennolla.

Toinen vaihtoehto on käyttää `noindentafter`²-pakettia ja määrittellä sen tarjoaman komennon avulla, minkä ympäristöjen jälkeen ei haluta sisennystä. Seuraava esimerkki poistaa sisennyksen aina `list-` ja `tabular-`ympäristöjen jälkeen (luvut 4.6.2 ja 4.7).

² <https://www.ctan.org/pkg/noindentafter>

```
1 \hangpara{2cm}{1}Tässä tekstikappaleessa on riippuva sisennys. Kappale
2 alkaa yhdellä sisentämättömällä rivillä, ja kappaleen seuraavat rivit
3 on sisennetty 2\,cm. Ei ole kovin vaikeaa.
```

⇒ Tässä tekstikappaleessa on riippuva sisennys. Kappale alkaa yhdellä sisentämättömällä rivillä, ja kappaleen seuraavat rivit on sisennetty 2 cm. Ei ole kovin vaikeaa.

Esimerkki 4.1: Riippuva sisennys `hanging`-paketin ja sen `\hangpara`-komennon avulla.

```
\NoIndentAfterEnv{list}
\NoIndentAfterEnv{tabular}
```

Tosin `noindentafter`-paketti ei ole aina toiminut luotettavasti yhdessä `polyglossia`-kielipaketin kanssa. Jos sisennyksen poistaminen ei tahdo toimia, kyse voi olla juuri tästä.

Kolmas keino kappaleen ensimmäisen rivin sisennyksen estämiseksi jonkin ympäristön jälkeen on se, että aloittaa tekstikappaleen heti ympäristön lopettavan `\end`-komennon jälkeen – ilman tyhjää riviä.

4.1.4 Riippuva sisennys

Riippuva sisennys tarkoittaa tekstikappaleen muotoa, jossa sisennetään kappaleen muita rivejä mutta ei ensimmäistä. Riippuvaa sisennystä käytetään esimerkiksi kirjallisuus- ja lähdeluetteloissa, joissa on tarpeellista saada henkilön nimi tai muu lähdemerkinnän hakusana erottumaan selvästi vasemmassa reunassa. Tämän oppaan lopussa sivulla 255 on esimerkki lähdemerkinnöistä.

Myös virallisten asiakirjojen muotoilussa käytetään riippuvaa sisennystä. Niissä kappaleen ensimmäinen rivi voi sisältää otsikon, joka on tasattu vasempaan reunaan. Otsikon perässä on sarkainhyppy tekstikappaleen sisennyksen tasalle, ja kappaleen muut rivit on sisennettyä samalla tasolle.

Helpoin tapa riippuvien sisennysten toteuttamiseen lienee `hanging`³-paketin käyttö. Paketti tuo uuden komennon `\hangpara`, jonka käyttöä esimerkki 4.1 selventää. Komennon ensimmäinen argumentti on sisen-

³ <https://www.ctan.org/pkg/hanging>

```

1 \hangpara{2cm}{1}\makebox[2cm][1]{Otsikko}Tässä tekstikappaleessa on
2 riippuva sisennys. Kappale alkaa yhdellä sientämättömällä rivillä,
3 joka sisältää näkymättömässä 2\,cm leveässä laatikossa olevan otsikon.
4 Kappaleen muut rivit on sisennetty 2\,cm.

```

⇒ Otsikko Tässä tekstikappaleessa on riippuva sisennys. Kappale alkaa yhdellä sientämättömällä rivillä, joka sisältää näkymättömässä 2 cm leveässä laatikossa olevan otsikon. Kappaleen muut rivit on sisennetty 2 cm.

Esimerkki 4.2: Asiakirjan tyylisten tekstikappaleiden toteutus.

nyksen mitta ja toinen argumentti määrittää, kuinka monta riviä kappaleen alusta jätetään sientämättä. Jos toinen argumentti on negatiivinen luku, on merkitys päinvastainen eli luvun itseisarvo määrittää, kuinka monta riviä kappaleen alusta sisennetään.

Komennon `\hangpara` vaihtoehtona on ympäristö `hangparas`, jonka sisällä kaikki kappaleet sisennetään riippuvalla tyyllillä samojen asetusten mukaisesti. Ympäristölle annetaan samat argumentit kuin `\hangpara`-komennollekin.

```

\begin{hangparas}{2cm}{1}
...
\end{hangparas}

```

Asiakirjan tyyllisen otsikon saa toteutettua `\makebox`-komennon avulla esimerkin 4.2 tavoin. Komento latoo näkymättömän laatikon, jonka leveys määritellään sisennyksen levyiseksi ja jonka sisään kirjoitetaan otsikko. Jos asiakirjatyylisiä tekstikappaleita tarvitaan useita, kannattaa määritellä sarkainleveyttä ja sisennystä varten oma mitta ja tekstikappaleen kirjoittamista varten oma komento. Seuraavassa on siitä esimerkki:

```

\newlength{\sarkain}
\setlength{\sarkain}{23mm}
\newcommand{\kappale}[1][\par\hangpara{2\sarkain}{1}%
\makebox[2\sarkain][1]{\ignorespaces #1}\ignorespaces}

```

Tämän jälkeen voi komennolla `\kappale` aloittaa asiakirjan sisennetyn tekstikappaleen. Komennolle voi antaa hakasulkeissa valinnaisen argu-


```

1 \begin{list}{}{
2   \setlength{\leftmargin}{2cm}
3   \setlength{\itemindent}{-2cm}
4 }
5 \item Tässä tekstikappaleessa on riippuva sisennys. Kappale alkaa
6 yhdellä sisentämättömällä rivillä, ja kappaleen muut rivit on
7 sisennetty 2\,cm.
8 \end{list}

```

Esimerkki 4.3: Riippuvan sisennyksen toteuttaminen `list`-ympäristön avulla.

mentin, joka on kappaleen otsikko. `\makebox`-komentoa ja muita laati-koita käsitellään tarkemmin luvussa 2.6.

Riippuvan sisennyksen voi toteuttaa myös `list`-ympäristön avulla. Se on tarkoitettu luettelien tekemiseen, mutta sopivilla asetuksilla yksi ”luotelman” kohta on riippuvasti sisennetty kappale. Tarkemmin `list`-ympäristöä käsitellään luettelien yhteydessä luvussa 4.6.2, mutta oheisessa esimerkissä 4.3 on sopivat asetukset riippuvan sisennyksen toteuttamiseen. Kappale alkaa `\item`-komennolla, koska kyseessä on ikään kuin luotelman kohta.

4.1.5 Vasen ja oikea sisennys sekä lohkolainaukset

Dokumentteihin tarvitaan välillä kokonaisia sisennettyjä tekstikappaleita, koska leipätekstin ohessa halutaan näyttää muuntyyppistä sisältöä. Kyse voi olla teksti- tai kuvaesimerkeistä, esimerkiksi muualta lainatusta tekstistä. Tässä oppaassa käytetään paljon sisennettyjä tekstikappaleita Latex-koodien esimerkkeihin.

Kokonaan sisennettyjä tekstikappaleita kutsutaan lohkolainauksiksi, koska ne ovat lainauksia, jotka käsittävät kokonaisen tekstilohkon. Lainausmerkkejä ei tarvitse käyttää, koska lainaus ilmaistaan typografisin keinoin. Sisennyksen lisäksi varsin yleistä on käyttää hieman pienempää kirjainleikkausta ja riviväliä kuin leipätekstissä. Joskus vasemman reunan sisennyksen lisäksi sisennetään myös oikeasta reunasta.

Latexissa on tavallisille lohkolainauksille kolme erilaista ympäristöä: `quotation`, `quote` ja `verse`. Kaksi ensin mainittua on tarkoitettu norma-

```

1 \newenvironment{lohkolainaus}{%
2   \begin{list}{}{
3     \setlength{\leftmargin}{1cm}
4     \setlength{\rightmargin}{1cm}
5     \setlength{\itemindent}{0bp}
6     \setlength{\listparindent}{\parindent}
7     \setlength{\parsep}{\parskip}
8     \setlength{\topsep}{1em}
9     \setlength{\partopsep}{0bp}
10  }
11  \item\linespread{1}\small
12 }{\end{list}}

```

Esimerkki 4.4: Lohkolainauksen eli tekstikappaleen vasemman ja oikean sisennyksen toteutus `list`-ympäristön avulla. Esimerkkikoodi määrittelee uuden ympäristön nimeltä `lohkolainaus`.

lilla tavalla juoksevalle tekstille, kun taas kolmas on tarkoitettu runon säkeiden ja säkeistöjen latomiseen.

`quotation`-ympäristö sisentää tekstikappaleiden ensimmäisen rivin 1,5 em:n verran, eikä kappaleiden välissä ole pystysuuntaista tilaa. `quote`-ympäristö ei sisennä kappaleiden ensimmäistä riviä, ja se puolestaan erottaa kappaleet toisistaan pystysuuntaisen tilan avulla. `verse`-ympäristöä käytetään siten, että lähdedokumentissa runon säkeet lopetetaan rivinvaihtokomentoon (`\`), lukuun ottamatta säkeistön viimeistä säettä. Säkeistöt erotetaan toisistaan tyhjällä rivillä, kuten Latexissa tekstikappaleet muutenkin. Lopputuloksena on useimpiin runoihin sopiva ladontatapa, jossa säkeistöjen vasen reuna on samalla tasolla, oikealla on liehureuna ja säkeistöjen välissä on pystysuuntaista tilaa.

Jos Latexin valmiit lohkolainausympäristöt eivät tuota haluttua lopputulosta, voi sisennetyt tekstikappaleet toteuttaa myös luettelien tekemiseen tarkoitetun `list`-ympäristön avulla (luku 4.6.2). Sopivilla asetuksilla ”luetelma” sisältää ihan tavallisen näköisiä tekstikappaleita, jotka vain on sisennetty vasemmalta tai oikealta tai molemmista reunoista.

Esimerkistä 4.4 selviää, kuinka `list`-ympäristöä voi käyttää sisennyksen toteuttamiseen. Esimerkki määrittelee uuden ympäristön nimeltä `lohkolainaus`, jota voi hyödyntää myöhemmin dokumentissa.

```

\begin{lohkolainaus}
  Tämä tekstikappale on sisennetty vasemmalta ja oikealta. Lisäksi
  kirjainleikkaus on hieman pienempi (\small) kuin leipätekstissä.
\end{lohkolainaus}

```

Omien ympäristöjen määrittelyä käsitellään tarkemmin luvussa 2.3. Esimerkin 4.4 rivillä 11 oleva `\item`-komento on pakollinen, koska se aloittaa `list`-ympäristöön kuuluvan luotelman kohdan. Sen perässä olevat komennot `\linespread` ja `\small` sen sijaan ovat vapaaehtoisia. Ne ovat mukana siksi, että on varsin tavallista latoa lohkolainaukset pienemmällä rivivälillä (rivikorkeudella) ja pienemmällä kirjainleikkauksella kuin leipäteksti.

4.1.6 Rivinvaihtokomennot

Latex-lähdedokumentissa olevat rivinvaihdot tulkitaan sanaväleiksi siinä missä välilyönnitkin, eli ne rivinvaihdot eivät päädy ladottuun dokumenttiin (luvut 2.1.2 ja 2.1.3). Sen sijaan ladottuun dokumenttiin saadaan rivinvaihto käyttämällä komentoa `\\` eli kaksi kenoviivaa. Komenton ei tarvitse sijaita lähdedokumentissa rivin lopussa.

```

ensimmäinen \\ toinen \\
kolmas

```

```

⇒ ensimmäinen
   toinen
   kolmas

```

Rivinvaihtokomennolle voi antaa hakasulkeissa valinnaisen argumentin, joka ilmaisee rivien väliin ladottavan ylimääräisen pystysuuntaisen tilan. Argumentin on siis oltava mitta.

```

ensimmäinen \\ toinen \\[1.3ex] kolmas

```

```

⇒ ensimmäinen
   toinen
   kolmas

```

Komennosta on olemassa tähtiversio `*`, joka edellisten ominaisuuksien lisäksi estää sivun vaihtumisen tämän rivinvaihdon kohdalla. Myös

tähtiversiolle voi antaa valinnaiseksi argumentiksi mitan, ja sen merkitys on sama kuin komennon normaalilla versiollakin.

Rivin voi vaihtaa myös komennolla `\newline`, mutta tämä komento ei hyväksy valinnaista argumenttia eikä siitä ole tähdellistä versiota. Komennot `\newline` ja `\\` käyttäytyvät eri tavoin taulukoissa, joita käsitellään luvussa 4.7.

4.1.7 Lesket ja orvot

Leski- ja orporivit tarkoittavat typografiassa rumannäköisiä yksinäisiä rivejä. Leskirivi (widow) on tekstikappaleen viimeinen rivi, joka on yksinään sivun tai palstan yläreunassa. Orporivi (orphan) puolestaan on tekstikappaleen ensimmäinen rivi, joka on yksin sivun tai palstan alareunassa. Molemmat voivat näyttää ikävältä, mutta yleensä orporivejä ei pidetä kovin vakavana virheenä; leskien välttämisessä ollaan enemmän tosissaan.

Ulkoasun lisäksi lesket ja orvot voivat olla ikäviä myös lukemisen kannalta. Kun tekstikappale vaihtuu, lukija pitää pienen tauon ja valmistautuu uuteen kappaleeseen. Sivun tai palstan olisi sopivaa vaihtua samassa kohdassa eli tekstikappaleiden välissä, mutta leski- ja orporivit aiheuttavat kaksi taukoa melkein peräkkäin: sekä kappaleiden välissä että sivun tai palstan vaihtumisen kohdalla.

Latexissa leski- tai orporivit lienee käytännöllisintä estää `nowidow`⁴-paketin avulla. Paketin lataamisen jälkeen käytetään komentoa `\setnowidow`, joka estää leskirivit eli pitää tekstikappaleen lopusta vähintään kaksi riviä yhdessä sivun tai palstan yläreunassa. Komennolle voi antaa hakasulkeissa valinnaisen argumentin (kokonaisluvun), joka ilmaisee, kuinka monta riviä täytyy vähintään pysyä yhdessä. Vastaavasti orporivit estetään komennolla `\setnoclub`, joka toimii samalla tavalla. Molemmat komennot vaikuttavat koko dokumenttiin eli kaikkiin tekstikappaleisiin.

```
\usepackage{nowidow}
\setnowidow % leskirivien esto
\setnoclub % orporivien esto
```

⁴ <https://www.ctan.org/pkg/nowidow>

Paketti `nowidow` määrittelee myös komennot `\nowidow` ja `\noclub`, joilla voi vaikuttaa yksittäisen tekstikappaleen leski- ja orporiveihin. Nämä komennot täytyy sijoittaa tekstikappaleen loppuun, ja myös niille voi antaa hakasulkeissa valinnaiseksi argumentiksi luvun, joka kertoo yhdessä pidettävien rivien määrän.

Jos ei halua tai voi käyttää `nowidow`-pakettia, voi leski- ja orporivit estää myös Texin matalan tason toimintojen avulla. Ladonta-algoritmi käyttää leski- ja orporiveissä sisäisesti haitallisuusarvoa tai sakkoarvoa (penalty), ja jos lesket ja orvot halutaan estää, määritellään niiden haitallisuusarvo mahdollisimman korkeaksi. Käytännössä arvo 10 000 tarkoittaa samaa kuin ääretön, eli silloin lesken tai orvon haitallisuus on niin suuri, ettei sellaisia sallita.

```
\widowpenalty 10000 % leskirivien esto
\clubpenalty 10000 % orporivien esto
```

Leskien ja orpojen haitallisuusarvoiksi voi kokeilla hieman pienempiäkin lukuja. Silloin leski- tai orporivit voidaan sallia joissakin tilanteissa, jos ladonta-algoritmi ei löydä parempaakaan ratkaisua.

Orvoksi kutsutaan myös tavua, joka jää yksin kappaleen viimeiselle riville. Se on häiritsevän näköinen ainakin silloin, kun tavu on kapeampi kuin seuraavan kappaleen ensimmäisen rivin sisennys. Orpotavujen estämiseen ei taida olla automaattisia keinoja, mutta kappaleen sanamäärää ja sanajärjestystä muuttamalla voi tietenkin vaikuttaa rivien latomiseen. Kappaleen viimeiseen sanaan voi kirjoittaa myös tavutusvihjeitä (`\-`) mutta jättää vihje pois ennen viimeistä tavua. Näin estetään lyhyen orpotavun muodostuminen. Tavutusvihjeitä ja muita tavutuksen asetuksia käsitellään luvussa 3.6.

4.1.8 Marginaali huomautukset

Klassisessa kirjatypografiassa – jota tämäkin opas suunnilleen noudattaa – on aukeaman ulkoreunoissa melko suuret marginaalit. Niitä voidaan käyttää eräänlaisina apupalstoina, joihin voi kirjoittaa lyhyehköjä huomautuksia ja lisätietoa.

Marginaali huomautukset tehdään komennolla `\marginpar`, jonka argumentiksi annetaan marginaaliin tuleva teksti. Marginaalin teksti la-

dotaan vasempaan tai oikeaan marginaaliin sen mukaan, kumpi on aukeaman ulkoreunassa. Yksipuolisilla sivuilla teksti ladotaan oletuksena oikeaan marginaaliin. Mikäli sivu on kaksipaltaisessa tilassa (luku 4.12), marginaalihuomautus ladotaan lähempänä olevaan marginaaliin.

```
Tämä on leipätekstin tekstikappale,  
\marginpar{Tämä ladotaan marginaaliin.}  
joka ladotaan sivun normaalille tekstialueelle.
```

Yleensä marginaalihuomautukset kannattaa latoa erilaisella kirjainleikkauksella kuin leipäteksti, jotta ne erottuvat toisistaan. Jos leipätekstissä käytetään antiikvaa, voisi marginaalissa käyttää esimerkiksi pienempää groteskia. Tämän toteutusta varten kannattaa määritellä uusi komento, vaikkapa nimellä `\huomautus`, jota käyttämällä marginaalihuomautukset saa helposti yhdenmukaiseksi. Seuraavassa on esimerkki tällaisen komennon määrittelystä:

```
\newcommand{\huomautus}[1]{%  
\marginpar{\sffamily\scriptsize\RaggedRight #1}}
```

Omien komentojen määrittelyä käsitellään tarkemmin luvussa 2.2 ja fonttiasetuksia luvussa 3.4. Edellä olevassa komennon määrittelyssä on mukana myös palstan muotoon vaikuttava komento `\RaggedRight`, josta on lisätietoa luvussa 4.1.1.

Tämän tekstikappaleen vieressä on esimerkki edellä mainitun tyylistä marginaalihuomautuksesta. Kirjainleikkauksen on syytä olla selvästi pienempi ja riittävän erilainen kuin leipätekstissä, jotta huomautus ei häiritse liikaa leipätekstin lukemista.

Sivun marginaalissa olevan huomautuspalstan leveys voidaan määrittää sivun asetusten yhteydessä. Asetuksia hoitavan `geometry`⁵-paketin valitsimella `marginparwidth` asetetaan palstan leveys ja valitsimella `marginparsep` palstan etäisyys leipätekstistä. Valitsimella `reversemarginpar` siirretään marginaalihuomautukset päinvastaiseen marginaaliin. Näitä ja muitakin sivun asetuksia käsitellään luvussa 3.2.

Edellä mainittuja asetuksia voi muuttaa myös kesken dokumentin. Mitta `\marginparwidth` määrittää marginaalihuomautuspalstan leveyden ja mitta `\marginparsep` sen etäisyyden leipätekstistä. Lisäksi on

⁵ <https://www.ctan.org/pkg/geometry>

Tässä on pienellä groteskilla ladottu huomautus.

mitta `\marginparpush`, jolla asetetaan peräkkäisten marginaalihuomautusten vähimmäisetäisyys toisistaan. Seuraavassa on esimerkki näiden mittojen asettamisesta:

```
\setlength{\marginparwidth}{50bp}
\setlength{\marginparsep}{10bp}
\setlength{\marginparpush}{6bp}
```

Kesken dokumentin komennolla `\reversemarginpar` voidaan vaihtaa marginaalihuomautukset vastakkaiseen marginaaliin ja komennolla `\normalmarginpar` palautetaan oletusasetukset.

4.1.9 Anfangit eli suurikokoiset alkukirjaimet

ANFANGIT ovat suurikokoisia tekstin alkukirjaimia. Tyypillisesti kirjaimen korkeus on kahdesta viiteen riviä ja se upotetaan kappaleen sisään. Joskus anfangi sijoitetaan kokonaan marginaalin puolelle tai se sijaitsee normaalisti tekstin peruslinjalla. Anfangikirjain voidaan latoa erilaisella kirjainleikkauksella kuin leipäteksti. Esimerkiksi vanhassa ja vanhantyyllisessä typografiassa anfangiin voi sopia erityisen koristeellinen kirjain.

EDELLÄ OLEVAAN, tähän ja seuraavaan tekstikappaleeseen anfangit sopivat huonosti, eikä niitä tavallisesti käytetä kirjan alalukujen alussa eikä väliotsikon jälkeen. Tässä ne ovatkin vain esimerkin vuoksi. Anfangien käytön varsinainen tarkoitus on osoittaa tekstin alkamiskohta esimerkiksi artikkelin alussa tai kirjan päälukujen alussa. Varsinkin aikakauslehden artikkelit sisältävät monenlaisia visuaalisia elementtejä kuten pää- ja alaotsikon, erillisen johdantokappaleen eli ingressin, kuvia ja tietolaatikoita. Anfangin avulla katse löytää artikkelin alun helposti.

KAPPALEEN ALOITTAVA SANA tai pidempikin ilmaus ladotaan joskus lihavoidulla kirjainleikkauksella tai pienversaalilla, kuten näissä esimerkeissä on tehty. Tällainen kappaleen aloitus voi toimia hieman otsikon tavoin tai sitten muuten vain ilmaisee uuden luvun alkamista.

Latexissa anfangien tekemistä varten on paketti `lettrine`⁶, joka tarjoaa käyttöön samannimisen komennon. Jokainen anfangi on erilainen, ja

⁶ <https://www.ctan.org/pkg/lettrine>

siksi `\lettrine`-komentokin sisältää useita asetuksia anfangin ulkoasun hienosäätöön. Komentoa käytetään seuraavalla tavalla:

```
\lettrine[lines=3, loversize=.06, lhang=.02, findent=-5bp,  
nindent=4bp, slope=4bp]{A}{nfangi} aloittaa tämän tekstikappaleen.
```

Edellä oleva esimerkki tekee kolmerivisen upotetun A-anfangin, jonka perässä oleva sanan osa ”nfangi” ladotaan pienversaalilla. Valitsimella `loversize` kasvatetaan hieman anfangikirjaimen kokoa, ja `lhang`-valitsimella siirretään kirjainta hieman vasemmalle marginaalin puolelle. Valitsin `findent` määrittää anfangikirjaimen rinnalla olevien rivien sisennyksen määrän suhteessa kirjaimen. `nindent` määrittää toisen ja sitä seuraavien rivien sisennyksen suhteessa ensimmäiseen riviin, ja `slope` määrittää sisennysportaan, joka kertautuu joka rivillä kolmannesta rivistä alkaen. Näillä asetuksilla saadaan A-kirjaimen viistoa reunaa mukailevat rivien alut.

Ennen `lettrine`-paketin käyttämistä kannattaa lukea paketin ohjekirjasta tietoa `\lettrine`-komennon valitsimista ja myös teknisistä rajoituksista. Anfangit eivät toimi Latexissa teknisesti joka paikassa – ja typografisesti vielä harvemmassa.

4.2 Tekstin korostaminen

Tekstin yksittäisten sanojen korostukset toteutetaan tavallisimmin siten, että käytetään leipätekstin kirjainperheestä jotakin poikkeavaa kirjainleikkausta, esimerkiksi *kursiivia*, *kallistettua*, **lihavoitua** tai PIEN-VERSAALIA. Joskus vaihdetaan koko kirjainperhe toiseksi, esimerkiksi ladotaan antiikvan sekaan groteskia tai tasalevyistä.

Korostukset siis liittyvät läheisesti fonttien valintaan ja asetuksiin, ja siksi korostusten teknistä toteuttamista ja Latex-komentoja käsitellään pääasiassa fonttien yhteydessä luvussa 3.4.3. Siellä muun muassa neuvotaan, kuinka kirjainperheet ja niiden eri kirjainleikkaukset valitaan. Sen sijaan nyt käsillä olevassa luvussa keskitytään tekstin korostamisen typografiaan käytäntöihin ja suosituksiin. Joitakin uusia komentojakin esitellään.

Tekstissä sanoja korostetaan siksi, että halutaan niiden erottuvan ympäristöstään. Tärkeän avainsanan pitää ehkä erottua tekstipalstasta

niin, että sen huomaa nopeasti tekstiä silmäilemällä. Se voi helpottaa tiedon löytämistä.

Sanoja tai ilmauksia korostetaan toisinaan myös siksi, että ne eivät ole kielen sanoja tavallisessa merkityksessään vaan lainauksia kielestä tai toisesta tekstityypistä. Ihmiskieltä käsittelevissä teksteissä on tavallista kursivoida sanat silloin, kun viitataan itse sanoihin eikä varsinaisesti käytetä kyseisiä sanoja. Sanoihin ja muuhun kieliainekseen viitatessa voi tosin käyttää myös lainausmerkkejä. Tietotekniikan koodikielen esimerkit ladotaan usein leipätekstistä poikkeavalla fontilla, jotta käy selväksi, että kyse on koodikielisestä lainauksesta. Tällöin hyvin tavallista on tasalevyisen fontin käyttö, koska se jäljittelee tekstieditorien ja komentotulkkia suorittavien pääteohjelmien fonttia.

Sopivan korostustavan valinnassa on olennaista saavuttaa riittävä kontrasti ympäröivään tekstiin. Liian lievä muutos ei erotu; toisaalta liian räikeä muutos voi olla häiritsevää. Yhteensopivat korostuskeinot saadaankin lähes aina saman kirjainperheen sisältä, eri leikkauksia käyttämällä.

4.2.1 Kursiivi, kallistus ja lihavointi

Antiikvatyyppisen kirjainleikkauksen korostamiseen sopii yleensä parhaiten saman perheen *kursiivi* (`\textit`). Lihavointi toimii huomattavasti huonommin antiikvan kanssa, koska pelkkä merkkien viivojen vahvistaminen ei tuo riittävä kontrastia.

⇒ Antiikvaperheen *kursiivileikkaus* tuo yleensä voimakkaan muutokontrastin ja on tyyllisesti yhteensopiva korostuskeino. Sen sijaan **lihavointi** erottuu antiikvassa huomattavasti huonommin.

Lihavointi (`\textbf`) toimii huomattavasti paremmin groteskityyppisen kirjainperheen kanssa, koska lihavoitu leikkaus on usein selvästi vahvempi normaaliin verrattuna. Groteskin kursiivi sen sijaan ei monesti edes ole varsinaisen kursivileikkaus vaan pelkkä lievä kallistus (`\textsl`). Sellainen ei välttämättä tuo riittävä kontrastia.⁷

⁷ Joissakin niin sanotuissa humanistisissa groteskiperheissä on mukana ihan käyttökelpoinen ja selvästi erottuva kursivileikkaus.

- ⇒ Groteskia on tavallista korostaa **lihavoinnilla** eli leikkauksella, joka on selvästi tavallista vahvempi. Sen sijaan *kursiivi* tai *kallistus* ei kaikissa groteskiperheissä erotu riittävästi.

Joissakin kirjainperheissä on mukana useita erivahvuisia leikkauksia. Silloin täytyy valita käyttöön riittävästi toisistaan poikkeavat vahvuudet. Esimerkiksi kaksi vierekkäistä vahvuutta eivät välttämättä poikkea toisistaan riittävästi, eikä selvää vahvuuskontrastia synny.

Mikäli leipätekstissä käytetään sulkeita ja niiden sisällä kursivoitua ilmausta (*kuten tässä*), ei itse suljemerkkejä pidä kursivoida, koska ne kuuluvat ulompaan tekstiin. Jos sulkeiden ulkopuolellakin on voimassa kursivointi, voi itse sulkeetkin kursivoida.

Latexissa kursiivin voi toteuttaa myös komennolla `\emph`, joka huomioi ympärillä olevan korostustilan. Tavallisesti komento kursivoi argumentiksi annetun tekstin, mutta jos kursivointi on jo päällä komennon ympärillä, se latookin argumentiksi annetun tekstin tavallisella pystyasentoisella kirjainleikkauksella. Komennon ajatuksena on, että voidaan samalla komennolla korostaa sekä pystyasentoisen tekstin että kursiivin keskellä, eikä tarvitse välittää tai tietää, kumpi leikkaus on voimassa.

```
tavallinen \emph{korostettu} tavallinen \\  
\emph{korostettu \emph{kaksoiskorostettu} korostettu}
```

- ⇒ tavallinen *korostettu* tavallinen
korostettu kaksoiskorostettu *korostettu*

4.2.2 Pienversaali eli kapiteeli

Melko rauhallinen typografinen keino on PIENVERSAALI eli kapiteeli (`\textsc`), jonka kirjaimet näyttävät versaalikirjaimilta mutta ovat suunnilleen gemenakirjainten eli pienten kirjainten korkuisia. Pienversaali on fonttiin sisältyvä erillinen suunnittelijan piirtämä leikkaus, eli se ei ole tietokoneen mekaanisesti pienentämä versio versaalikirjaimista.

Jos fonttiin ei sisälly pienversaalia ja sellaisen silti haluaa omaan keinovalikoimaansa, voi yrittää vähemmän tyylikästä mekaanista toteutusta luvussa 3.4.11 kerrotuilla asetuksilla. Luvuissa 3.4.9 ja 4.2.4 puolestaan käsitellään merkkivälien `HARVENNUSTA`, joka voi maltillisesti käytettynä sopia pienversaaleihin.

Esimerkit	Kirjainten ja numeroiden muoto
Väärin: ISO-8859-1 2012 a.D.	versaalikirjaimet ja gemenanumerot
Oikein: ISO-8859-1 2012 A.D.	pienversaalikirjaimet ja gemenanumerot
Oikein: ISO-8859-1 2012 A.D.	versaalikirjaimet ja versaalinumeroit

Taulukko 4.2: Versaalikirjainten, pienversaalien sekä gemenanumeroiden vertailua. Versaalikirjaimet ja gemenanumerot eivät sovi yhteen.

Korostustarkoituksessa pienversaalialia käytetään esimerkiksi pienissä väliotsikoissa, koska sen avulla syntyy selvä muotokontrasti leipätekstin kanssa. Toisinaan tekstikappaleen ensimmäisiä sanoja ladotaan pienversaalilla, jolloin se toimii otsikon kaltaisessa tehtävässä. Myös kirjallisuus- ja lähdeluetteloissa käytetään välillä pienversaalialia tekijöiden nimissä, koska se helpottaa nimen mukaan aakkostetun luettelon silmäilyä.

Pienversaalialia käytetään toisinaan versaalikirjainten sijasta. Silloin tarkoituksena ei ole tekstin korostamiseen vaan päinvastoin häiritsevän korostuksen välttäminen. Esimerkiksi versaalikirjaimiset lyhenteet (SPR, HTML) voivat paljon käytettynä erottua leipätekstistä ikävän selvästi, eikä tekstipalsta näytä tasaiselta. Pienversaalit sulautuvat muihin tekstiin paremmin (SPR, HTML).

Typografisesta estetiikasta juontuu sekin, että pienversaalialia on sopiva pari gemenanumeroiden kanssa erilaisiin teknisiin koodeihin ja muihin ilmauksiin, jotka koostuvat kirjaimista ja numeroista. Sen sijaan versaalikirjaimia ja gemenanumeroita ei sovi käyttää yhdessä, koska merkien kokoero on häiritsevän suuri. Kuten taulukosta 4.2 näkyy, versaalit ovat joskus miltei kaksinkertaisia gemenanumeroihin verrattuna, ja muutenkin gemenanumerot asettuvat keskimäärin eri korkeustasolle.

Fontin näkökulmasta gemenanumerot ovat erillinen ominaisuus, jonka fontin suunnittelija on toteuttanut. Ominaisuuden käyttöönottoa ja muita numeroihin liittyviä fonttiasetuksia käsitellään luvussa 3.4.8.

4.2.3 Alleviivaus ja yliviivaus

Alleviivaus on käsin kirjoitetun tekstin ja mekaanisten kirjoituskoneiden korostuskeino, mutta nykyajan typografiaan se ei oikeastaan sovi. Alleviivaus nimittäin vaikeuttaa kirjainhahmojen tunnistamista ja hidastaa lukemista. Parempia korostuskeinoja ovat esimerkiksi kursiivi ja lihavointi. Alleviivaus voi kuitenkin joskus olla kätevä keino ohjata lukijan huomio tiettyyn kohtaan esimerkiksi pidemmässä esimerkkitekstissä.

Latexin oma alleviivauskomento `\underline` piirtää viivan tekstin tai muun argumentiksi annetun sisällön alapuolelle. Komennon huono puoli on, että erilliset alleviivaukset eivät välttämättä osu samalle korkeudelle. Seuraavassa esimerkissä toisen sanan alleviivaus on alempana j- ja y-kirjainten alapidennyksen vuoksi.

```
\underline{kone}, \underline{öljy}
⇒ kone, öljy
```

Toinen `\underline`-komennon ongelma on, että se estää sanojen tavuttamisen ja muutenkin tekstin katkaisemisen rivin lopussa. Komennon argumentiksi annettua tekstiä ei katkaista edes sanavälien kohdalta. Komennon hyvä puoli on, että se toimii myös matematiikkatilassa (luku 4.16).

Käytännössä alleviivaukset on parempi toteuttaa esimerkiksi paketin `ulem`⁸ avulla. Tavallisen alleviivauksen lisäksi se sisältää kaksoisalleviivauksen, aaltoviivat, pisteviivat ja erilaisia ylikirjoitusmerkintöjä. Taulukkoon 4.3 on koottu paketin sisältämiä komentoja.

Oletuksena `ulem`-paketti määrittelee Latexin `\emph`-korostuskomennon (luku 4.2.1) uudelleen siten, että komento tuottaa kursiivin sijasta alleviivattua tekstiä. Ajatuksena lienee se, että paketin avulla voi helposti muuttaa typografisen peruskorostusten eli kursiivin tilalle kirjoituskoneen korostustavan eli alleviivauksen. Käytännössä tällaista tuskin halutaan koskaan, ja onneksi paketin lataamisen yhteydessä voi käyttää valitsinta `normalem`, joka estää Latexin komentojen uudelleen määrittelyn.

⁸ <https://www.ctan.org/pkg/ulem>

Komento	Merkitys	Komento	Merkitys
<code>\uline</code>	alleviivaus	<code>\uuline</code>	<u>kaksoisalleviivaus</u>
<code>\uwave</code>	aaltoviiva	<code>\sout</code>	yliviivaus
<code>\dashuline</code>	katkoviiva	<code>\xout</code>	poisto
<code>\dotuline</code>	pisteviiva		

Taulukko 4.3: Alleviivaus- ja ylikirjoitusmerkitöjä `ulem`-paketin komentojen avulla.

```
\usepackage[normalem]{ulem}
```

Paketin `ulem` myötä on käytettävissä mitta `\ULdepth`, joka on alleviivauksen etäisyys tekstin peruslinjasta. Mitta asetetaan `\setlength`-komennolla (luku 2.4). Viivan paksuus puolestaan sisältyy komennon `\ULthickness` määritelmään, ja komennon voi halutessaan määrittellä uudelleen. Seuraavassa on näistä esimerkki:

```
\setlength{\ULdepth}{.2ex} % viivan etäisyys
\renewcommand{\ULthickness}{.1ex} % viivan paksuus
\uline{kone}, \uline{öljy}
```

⇒ kone, öljy

Toinen paketti alleviivausten ja yliviivausten tekemiseen on `soul`⁹. Se ei sisällä erikoisempia korostustapoja kuten aalto- ja katkoviivoja, mutta viivan etäisyyden ja paksuuden määrittämiseen on hieman kätevämmät komennot.

Tekstin alleviivauksen lisäksi on olemassa pari komentoa, joilla saa ladottua pelkän viivan ilman tekstiä. Komento `\hrulefill` latoo kaiken tilan täyttävän alaviivan ja komento `\dotfill` kaiken tilan täyttävän pisteviivan. Myös `\rule`-komento voi soveltua alaviivojen toteuttamiseen. Komentoa käsitellään erikoismerkkien yhteydessä luvussa 2.1.14.

```
abc \hrulefill\ abc \dotfill\ abc \rule[-.3ex]{5em}{.4bp} abc
```

⇒ abc _____ abc abc _____ abc

⁹ <https://www.ctan.org/pkg/soul>

4.2.4 Harvennus ja tiivistys

Merkkivälien harventaminen on tyyli- tai korostuskeino, joka juontuu antiikin roomalaisten kiveen hakatuista teksteistä. Historiansa vuoksi harvennus sopii luontevimmin antiikvan eli pääteviivallisen kirjainperheen versaalili- tai pienversaalikirjainten (luku 4.2.2) kanssa. Antiikin aikana gemenakirjaimet eivät olleet vielä käytössä. Tässä alaluvussa keskitytään harventamisen ja tiivistämisen typografiaan; teknisiä ohjeita on luvussa 3.4.9.

```
\large TAVALLINEN \\  
{\addfontfeatures{LetterSpace=10} KLASSINEN HARVENNUS}
```

⇒ TAVALLINEN
KLASSINEN HARVENNUS

Klassisesti eli antiikin tyyliin harvennetut (n. 10 %) versaalikirjaimet voivat sopia joihinkin otsikoihin. Sen sijaan pienversaalit lievästi HARVENNETTUNA (esim. 6 %) voivat olla tyylikäs tehostuskeino leipätekstiin.

Edellisiä esimerkkejä suurempi harvennus on huomiota herättävä korostus- tai tehokeino, joka voi sopia yksittäisiin sanoihin leipätekstin ulkopuolella. Jos kirjaimet ovat hyvinkin kaukana toisistaan, täytyy sanan ympärillä olla runsaasti tyhjää tilaa, jotta sanahahmo pysyy visuaalisesti koossa.

```
{\sffamily\bfseries\addfontfeatures{LetterSpace=40} SEIS!}
```

⇒ S E I S !

Merkkivälien tiivistäminen on joskus tarpeen suurikokoisissa otsikoissa. Fontin merkkivälit on ehkä suunniteltu ensisijaisesti pienten kookojen ehdoilla, ja siksi reilusti suuremmilla fonttiko'illa välit voivat olla häiritsevän suuria. Erityisesti tämä pätee groteskiin eli pääteviivattomaan kirjainperheeseen, kuten kuva 4.1 osoittaa. Antiikvat eivät yleensä siedä tiivistämistä, koska kirjainten pääteviivat asettuvat helposti päällekkäin.

Tavallinen Tiivistetty

Kuva 4.1: Otsikoiden merkkivälien tiivistäminen voi olla tarpeen suurikokoisen groteskin kanssa – varsinkin jos lihavointi ei ole käytössä.

4.2.5 Tietokonekoodi ja tasalevyinen kirjainperhe

Tietokonekoodi tai muu vastaava tekninen, koneelle syötettävä tai koneen tulostama koodi on tapana esittää tasalevyisellä fontilla. Kyseessä on ikään kuin lainaus toisenlaisesta tekstityypistä, joka ei ole luonnollista kieltä lainkaan.

Tasalevyinen kirjainperhe määritellään fonttiasetusten yhteydessä (luku 3.4.1) ja valitaan käyttöön esimerkiksi komennolla `\texttt`. Tasalevyisellä fontilla ladottava teksti on tavallisesti tietokonekoodia, joten sille on asetettu hieman toisenlaiset oletusasetukset kuin muille kirjainperheille. Oletuksena tekstiä ei tavuteta, ja esimerkiksi `\texttt`-komennon argumentiksi annettu teksti katkaistaan vain sanavälien ja yhdysmerkkien kohdalta. Myös niin sanotut Tex-ligatuurit (esim. ' ja --) on kytketty pois päältä. Fonttien oletusasetuksia ja niiden muuttamista käsitellään luvussa 3.4.6. Tex-ligatuureja eli Texin merkintätapoja lainausmerkeille ja ajatusviivoille käsitellään luvuissa 2.1.9 ja 2.1.10.

Toinen vaihtoehto tasalevyisen tekstin tuottamiseen on `\verb`-komento. Komennon nimi on lyhennetty englannin kielen sanasta *verbatim*, joka tarkoittaa 'kirjaimellisesti' tai 'sananmukaisesti'. Niin se juuri toimiikin, eli `\verb`-komennolle annettu argumentti tulkitaan kirjaimellisesti. Tex-järjestelmän omat merkintätavat kuten kenoviivalla (\) alkavat komennot eivät toimi tämän komennon argumentissa.

`\verb`-komennon argumentti täytyy ilmaista tavallisesta poikkeavalla tavalla. Välittömästi komennon nimen jälkeen tuleva merkki on oleva

argumentin aloitus- ja lopetusmerkki. Niiden välissä oleva teksti tulkitaan kirjaimellisesti ja ladotaan tasalevyisellä kirjainperheellä. Seuraavassa on tästä kaksi esimerkkiä:

```
\verb.tasalevyinen. % aloitus- ja lopetusmerkinä piste
\verb|tasalevyinen| % aloitus- ja lopetusmerkinä pystyviiva
```

⇒ tasalevyinen tasalevyinen

Yksittäisiä sanoja pidempien tietokonekoodien latomiseen on olemassa `verbatim`-ympäristö, jonka sisällä kaikki teksti tulkitaan kirjaimellisesti – myös välilyönnit ja rivinvaihdot.

```
\begin{verbatim}
tieto-     kone-
          koodia
\end{verbatim}
```

⇒ tieto- kone-
 koodia

Paljon monipuolisemman ympäristön tarjoaa paketti `fancyvrb`¹⁰. Sen mukana tuleva `verbatim`-ympäristö ymmärtää monenlaisia asetuksia, joilla vaikutetaan tietokonekoodin latomiseen. Koodiin saa mukaan esimerkiksi rivinumerot, tai sen voi latoa kehyksen sisään. Myös koodin sisennykseen, fontin kokoon ja väreihin voi vaikuttaa.

4.2.6 Verkko-osoitteet

Verkko-osoitteiden eli internetin linkkien latominen voi olla pienoinen typografinen ongelma: osoitteet ovat pitkiä, eivätkä ne sisällä välilyönnejä, joista ne voisi luontevasti katkaista rivin vaihtumisen kohdalla. Tavuviivojen lisääminen rivin loppuun voi olla ongelmallista, koska kyse on teknisestä merkkijonosta, joka pitäisi tulkita kirjaimellisesti.

Parasta olisi latoa verkko-osoitteet leipäteksti ulkopuolelle, esimerkiksi sivun alareunaan alaviitteeksi (luku 4.11). Tällä tavoin pitkät ja joskus rumat osoitteet eivät häiritse leipätekstin lukemista mutta ovat silti lukijan saatavilla.

¹⁰ <https://www.ctan.org/pkg/fancyvrb>

Latexissa verkko-osoitteet kannattaa toteuttaa siihen tarkoitettuun pakettiin avulla. Paketti `hyperref`¹¹ sisältää toimintoja paitsi verkko-osoitteiden latomiseen mutta myös pdf-tiedoston asetusten (luku 3.3) ja ristiiviitteiden (luku 4.10) hallintaan. Paketin avulla valmiin pdf-tiedoston verkkolinkit toimivat, eli pdf-lukijassa linkkiä napsauttamalla pitäisi avautua verkkoselain ja linkin osoittama sivu.

Paketin käyttämisen typografinen hyöty on se, että rivin vaihdon kohdalle sattuvat verkko-osoitteet katkaistaan automaattisesti luontevista kohdista eli vain tiettyjen välimerkkien jälkeen. Tavuviivoja ei lisätä rivin loppuun. Myös typografiset ligatuurit (luku 3.4.7) kytketään pois päältä.

`hyperref`-paketin ohjekirja neuvoo lataamaan paketin viimeisenä eli muiden pakettien lataamisen jälkeen, koska se muokkaa joidenkin muualla määriteltyjen komentojen ominaisuuksia.

```
\usepackage{hyperref} % paketin lataaminen
\hypersetup{hidelinks} % linkeistä pois kehukset
\urlstyle{same} % linkkien fontti
```

Edellisen esimerkin komennolla `\hypersetup` vaikutetaan paketin asetuksiin. Oletuksena esimerkiksi linkkien ympärille ladotaan värilliset kehukset, mutta valitsin `hidelinks` poistaa ne. Valitsimella `urlcolor` voisi vaihtaa verkko-osoitteiden kehuksen väriä.

Esimerkin seuraava komento `\urlstyle` määrittää, mitä kirjainperhettä käytetään verkko-osoitteiden latomiseen. Komennon argumentti `same` tarkoittaa, että käytetään samaa kirjainperhettä kuin ympärillä olevassa tekstissä. Tietty kiinteä kirjainperhe valitaan `\urlstyle`-komennon argumentilla `rm` (antiikva, roman), `sf` (groteski, sans serif) tai `tt` (tasalevyinen). Oletusasetus on tasalevyinen kirjainperhe.

Verkko-osoitteiden ilmaisemisen peruskomento on `\href`, jolle annetaan kaksi pakollista argumenttia: ensimmäinen on varsinainen osoite ja toinen on dokumenttiin ladottava teksti. Linkin teksti voi siis olla mitä hyvänsä tekstiä, ja pdf-lukijassa tekstiä napsauttamalla linkin osoittama sivusto avautuu.

```
\href{https://www.ctan.org/}{CTAN-sivusto}
```

¹¹ <https://www.ctan.org/pkg/hyperref>

⇒ CTAN-sivusto

Kun halutaan ladata dokumenttiin itse verkko-osoite, käytetään komentoa `\url`, joka osaa katkaista osoitteet järkevällä tavalla rivin lopussa. Komennolle annetaan vain yksi argumentti, verkko-osoite.

```
\url{https://www.ctan.org/}
```

⇒ <https://www.ctan.org/>

Joskus dokumenttiin voidaan tarvita verkko-osoite, joka ei ole pdf:ssä napsautettava linkki. Silloin käytetään komentoa `\nolinkurl`. Tämän komennon argumenttina oleva verkko-osoite ainoastaan ladotaan osoitteiden tavoin.

Linkkinä toimivien sähköpostiosoitteiden latominen vaatii `\href`- ja `\nolinkurl`-komentojen yhdistämistä. Linkin alkuun täytyy lisätä sana `mailto:`, mutta tuota sanaa tuskin halutaan näkyviin itse dokumenttiin. Niinpä sähköpostiosoite kannattaa kirjoittaa seuraavalla tavalla:

```
\href{mailto:tunnus@osoite.net}{\nolinkurl{tunnus@osoite.net}}
```

Osoite pitää siis kirjoittaa kaksi kertaa: itse linkkiä varten ja dokumenttiin ladottavaa tekstiä varten. Oma työtä kannattaa helpottaa määrittelemällä sähköpostiosoitteita varten oma komento, esimerkiksi seuraavalla tavalla:

```
\newcommand{\sposti}[1]{\href{mailto:#1}{\nolinkurl{#1}}}
```






4.2.7 Värit

Eri värien käyttö voi sopia tekstin erilaisten osien korostamiseen. Esimerkiksi otsikot, taulukot tai leijuvat osat saa erottumaan selvemmin, kun niiden toteutuksessa käyttää värejä. Sen sijaan leipätekstin sisällä värejä täytynee käyttää hillitysti, sillä värikäs tekstipalsta näyttää rauhattomalta ja voi hidastaa lukemista.

Latexiin saa värit `xcolor`¹²-paketin avulla.¹³ Se tarjoaa muun muassa komentoja, joilla tekstin väriä, tekstin taustaväriä tai sivun taustaväriä

¹² <https://www.ctan.org/pkg/xcolor>

¹³ Toinen vaihtoehto on yksinkertaisempi väripaketti `color`, joka sisältää värien käsittelyn perustoiminnot.

Malli	Parametrit	Esimerkki
rgb	punainen, vihreä, sininen	1, .7, .3 
HTML	punainen vihreä sininen	33aaaf3 
cmyk	syaani, magenta, keltainen, musta	.1, 1, .5, 0 
hsb	sävy, kylläisyys, kirkkaus	.2, 1, .6 
gray	harmaa	.55 

Taulukko 4.4: Erilaisia värimalleja ja niiden parametreja `xcolor`-paketissa.

voi muuttaa. Lisäksi se sisältää värien sekoittamiseen ja muuntamiseen liittyvää tekniikkaa. Paketin komentojen avulla voi myös värittää taulukoiden (luku 4.7) rivit vuorottelevin värein.

Omassa dokumentissa käytettävät värit on hyvä määritellä alussa. Määrittely tarkoittaa sitä, että annetaan väreille nimet – esimerkiksi käyttötarkoituksen mukaan – ja myöhemmin lähdedokumentissa viitataan väreihin käyttämällä alussa määriteltyjä nimiä. Näin väri on määritelty yhdessä paikassa ja sen muuttaminen on myöhemmin helppoa. On kyllä mahdollista käyttää värejä ilman etukäteismäärittelyäkin.

Omat värit määritellään komennolla `\definecolor`. Se tarvitsee kolme argumenttia: värin nimen, värimallin ja värikoodin. Komennon argumenttien rakenne on seuraavanlainen:

```
\definecolor{nimi}{välimalli}{parametrit}
```

Komennon argumentti `nimi` on kirjoittajan itse valitsema nimi, ja se voi liittyä esimerkiksi värin käyttötarkoitukseen (”nuoli”) tai värin yleiseen nimeen (”keltainen”). Argumentti `värimalli` tarkoittaa värin ilmaisemisen tapaa. Niitä on useita erilaisia, mutta yleisimmät on koottu taulukon 4.4.

Eri värimalleilla on eri määrä parametreja, joilla ilmaistaan värin ominaisuuksia. Kunkin parametrin arvo on yleensä desimaaliluku 0–1, mutta HTML-värimallissa parametrit ilmaistaan yhtenä kolmen kaksinumeroisen heksadesimaaliluvun sarjana HTML-merkintäkielen tavoin. Värin parametrit kirjoitetaan komennon argumenttiin `parametrit` ja ne erotetaan toisistaan pilkulla. Taulukon 4.4 Esimerkki-sarakkeessa on parametrien antamisesta esimerkki. Muista värimalleista voi lukea `xcolor`-paketin ohjekirjasta.

Kun värit on määritelty, niitä voi käyttää eri komentojen yhteydessä. Tekstin väri vaihdetaan toiseksi komennolla `\color`, josta on kaksi eri-laista versiota. Komento vaihtaa värin pysyvästi nykyisen ympäristön (luku 2.3) tai aaltosulkeilla (luku 2.1.6) rajatun alueen sisällä.

```
\color{nimi} % väriksi ennalta määritelty "nimi"  
\color[värimalli]{parametrit} % värimallin ja parametrien avulla
```

Toinen vaihtoehto on käyttää komentoa `\textcolor`, jolle annetaan edelliseen verrattuna vielä yksi argumentti lisää. Argumenttina on teksti, johon värin halutaan vaikuttavan. Vaikutusalueena on siis vain kyseinen teksti.

```
\textcolor{nimi}{teksti}  
\textcolor[värimalli]{parametrit}{teksti}
```

Tekstin taustaväri vaihdetaan komennolla `\colorbox` ja `\fcolorbox`. Komennot luovat laatikon, jonka sisällä taustaväri on voimassa. Ensin mainittu komento vaihtaa vain taustavärin, ja jälkimmäinen sisältää lisäksi värillisen kehyksen, eli sille määritellään kaksi väriä.

```
\colorbox{nimi}{teksti} % taustaväriksi ennalta määritelty "nimi"  
\colorbox[värimalli]{parametrit}{teksti}  
\fcolorbox{nimi}{kehyksen nimi}{tausta}{teksti}  
\fcolorbox[värimalli]{parametrit}{kehyksen nimi}{parametrit}{tausta}{teksti}
```

Kehystetyn laatikon kehysviivan leveys on mitassa `\fboxrule` ja kehyksen etäisyys sisällöstä mitassa `\fboxsep`. Seuraavassa on esimerkki useiden edellä mainittujen komentojen käytöstä. Tästä tuskin kannattaa ottaa mallia vakavaan typografiseen työhön.

```
\definecolor{pun}{rgb}{1,.2,.2}  
\setlength{\fboxrule}{2bp} \setlength{\fboxsep}{1bp}  
\fcolorbox[gray]{.4}{.2}{\color[gray]{1}Tekstiä}  
\textcolor{pun}{voi korostaa} \colorbox{pun}{väreillä}.
```

⇒ **Tekstiä** voi korostaa väreillä.

Sivulle voi määrittää taustavärin komennolla `\pagecolor`. Taustaväri vaihtuu nykyiselle ja seuraaville sivuille, kunnes väri vaihdetaan toiseksi tai poistetaan. Komennolle annetaan samanlaiset argumentit kuin muillekin värikomennolle. Komento `\nopagecolor` poistaa taustavärin kokonaan eli palauttaa taustan oletustilaan.

```
\pagecolor{nimi} % sivun taustaväriksi ennalta määritelty "nimi"  
\pagecolor{värimalli}{parametrit}  
\nopagecolor
```

4.3 Sivunvaihdot ja sivujen tasaaminen

Kirjoittajan ei normaalisti tarvitse huolehtia sivun vaihtumisesta lainkaan, vaan järjestelmä pyrkii latomaan hyvännäköisiä sivuja automaattisesti. Välillä tietenkin halutaan aloittaa jokin uusi aihe puhtaalta sivulta, ja sitä varten Latexissa on muutama komento. Mainittakoon kuitenkin heti aluksi, että otsikoiden yhteyteen saa sivunvaihdon kätevimmin keinoilla, jotka neuvotaan otsikoiden yhteydessä luvussa 4.4.4.

Komento `\clearpage` vaihtaa sivua, eli komennon jälkeinen dokumentin osa alkaa aina puhtaalta sivulta. Lähes samanlainen komento on `\cleardoublepage`, joka kaksipuolisessa dokumentissa aloittaa uuden sisällön aina parittomilta sivuilta. Jos seuraavana ei satu olemaan pariton sivu, komento tekee väliin tyhjän sivun. Tätä ominaisuutta on käytetty tässä oppaassa: pääluvut alkavat aina oikeanpuoleiselta eli parittomalta sivulta.

`\cleardoublepage`-komennon aiheuttamalle tyhjälle sivulle saatetaan kuitenkin latoa ylä- ja alatunnisteet, jos sellaiset on määritelty (luku 3.2.4). Täysin tyhjän sivun saa käyttämällä apuna pakettia `titlesec`¹⁴ ja sen valitsinta `cleareempty`:

```
\usepackage[cleareempty]{titlesec}
```

Edellä mainittu `titlesec`-paketti sisältää monenlaisia toimintoja otsikoiden muotoiluun, joten sitä käsitellään tarkemmin otsikoiden yhteydessä luvussa 4.4.4.

Komennoilla `\clearpage` ja `\cleardoublepage` on lisäksi sellainen ominaisuus, että ne pakottavat aiemmin määritellyt leijuvat osat (luku 4.9) ladottavaksi ennen uutta puhdasta sivua.

Hieman toisenlainen sivunvaihtokomento on `\newpage`. Se sallii aiempien leijuvien osien latomisen myös sivunvaihdon jälkeen. `\newpage`

¹⁴ <https://www.ctan.org/pkg/titlesec>

eroaa edellisistä sivuvaihtokomennoista myös siten, että kaksipalstaisessa tilassa (luku 4.12) se aloittaa vain uuden palstan, ei välttämättä uutta sivua.

Sivunvaihdon todennäköisyyteen tietyssä kohdassa voi vaikuttaa komennoilla `\pagebreak` ja `\nopagebreak`. Kummallekin voi antaa hakasulkeissa valinnaisen argumentin, joka on luku 0–4. Nolla tarkoittaa pienintä todennäköisyyttä sivunvaihdolle (`\pagebreak[0]`) tai sen välttämiseksi (`\nopagebreak[0]`). Luku neljä (joka on oletusarvo) tarkoittaa, että sivunvaihto pakotetaan tai estetään siinä kohdassa. Näiden komentojen tarkoitus on kertoa ladonta-algoritmille vihje, kuinka toivottavaa sivunvaihto on tietyssä kohdassa. Jos ei käytä suurinta arvoa neljä (4), algoritmi voi löytää omasta mielestään paremmankin sivunvaihtokohdan.

```
\pagebreak[3] % Lisätään sivunvaihdon todennäköisyyttä.  
\nopagebreak % Estetään sivunvaihto (oletusarvo 4).
```

Jos `\pagebreak`-komentoa käyttää tekstikappaleen sisällä, sivunvaihto ei välttämättä satu juuri komennon kohdalle. Sivun vaihdetaan kohdalle osuvan rivin lopussa.

Oletuksena Latex pyrkii latomaan kaikkien sivujen ylä- ja alareunan tasaiseksi. Toisin sanoen se saattaa venyttää tai kutistaa sivulla olevia pystysuuntaisia välejä niin, että kaikki sivut näyttävät yhtä korkeilta. Tämä tila voidaan asettaa päälle komennolla `\flushbottom`, mutta se on jo oletuksena päällä ainakin Latexin perusdokumenttiluokissa.

Vaihtoehtoinen tila asetetaan komennolla `\raggedbottom`. Komennon seurauksena sivujen alareunoja ei tasata. Toisin sanoen sivun pystysuuntaisiin väleihin ei kosketa, joten eri sivujen alareunat voivat sattuakin eri korkeudelle.

Dokumentin viimeistelyvaiheessa sivujen hienosäätöön sopinee komento `\enlargethispage`, jolla voi muuttaa käsillä olevan sivun korkeutta, käytännössä `\textheight`-mittaa. Komennolle annetaan argumentiksi positiivinen tai negatiivinen mitta. Seuraavassa on esimerkkejä:

```
\enlargethispage{12bp} % 12 pistettä korkeemmaksi  
\enlargethispage{-4bp} % 4 pistettä matalammaksi  
\enlargethispage{\baselineskip} % yksi tekstirivi lisää
```

Komennon tähtiversio `\enlargethispage*` pyrkii lisäksi tiivistämään sivun sisältöä pystysuunnassa niin paljon kuin mahdollista. Sen tyypillisin käyttötarkoitus lienee se, että sivulle pyritään mahduttamaan yksi ylimääräinen rivi. Tämäntyyppinen sivujen hienosäätö kannattaa tehdä vasta viimeisenä eli siinä vaiheessa, kun kaikki sisältö on valmiina ja muu typografinen asettelu tehty.

4.4 Jäsennys

Tekstin jäsentäminen tarkoittaa sisällön järjestystä ja sen jakamista sopiviin kokonaisuuksiin. Esimerkiksi laaja teos voi koostua osista, osien sisäisistä pääluvuista ja niiden alaluvuista. Eri luvut ilmaistaan eritasoisilla otsikoilla. Varsinaisten sisältölukujen jälkeen teoksessa voi olla erilaisia liite- tai luettelosivuja, esimerkiksi lähdeluettelo ja asiahakemisto. Toisaalta lyhyempi artikkeli sisältää ehkä vain yhden- tai kahdentasoisia otsikoita – muuta ei tarvita. Tässä alaluvussa käsitellään juuri tämäntyyppisiä asioita eli tekstikappaletta suurempien kokonaisuuksien jäsentämistä.

4.4.1 Perustiedot, nimiösivu ja tiivistelmä

Latexissa on yksinkertainen toiminto dokumentin perustietojen latomiseen dokumentin alkuun, mahdollisesti erilliselle nimiösivulle. Perustietoja ovat teoksen nimi, tekijä ja päiväys. Lähdedokumentin alussa perustiedot ilmaistaan seuraavilla komennoilla:

```
\title{Tieteellistä tiedettä}           % teoksen nimi
\author{Timo Tiedemies \and Tuija Tutkija} % tekijöiden nimet
\date{\today}                          % päiväys
```

Edellisen esimerkin kommentojen argumentissa voi käyttää komentoa `\thanks`, joka on tarkoitettu kiitosten osoittamiseen. Komennolle annetaan argumentiksi lyhyehkö teksti eli kiitosviesti, joka tulisi näkymään sivulla alaviitteessä.

Komennot eivät itse vielä lado mitään tekstiä dokumenttiin. Ne vain tallentavat perustiedot muistiin mahdollista myöhempää käyttöä varten. Komennolla `\maketitle` voi ladata kaikki edellä mainitut tiedot. Ne

voivat näkyä ensimmäisen sisältösivun alussa tai erillisellä nimiösivulla; se riippuu dokumenttiluokan asetuksista ja valitsimista `titlepage` ja `notitlepage`. Se, kumpi asetus on oletuksena päällä, vaihtelee eri dokumenttiluokissa (luku 3.1.2). Yksittäiset perustiedot saa ladottua seuraavilla komennoilla: `\thetitle`, `\theauthor` ja `\thedate`.

Perustietokomentoihin `\title`, `\author` ja `\date` ei ole tarkoitus sisällyttää kovin monimutkaisia muotoilukomentoja. Jos haluaa vaikuttaa perustietojen ulkoasuun, kannattaa käyttää siihen tarkoitettua pakettia `titling`¹⁵. Toisaalta mikään ei estä muotoilemasta oman dokumentin perustietoja ilman Latexin valmiita keinoja.

Hyödyllisiä ympäristöjä dokumentin alussa käytettäväksi voivat olla `titlepage` ja `abstract`. Ensin mainittu luo tyhjän sivun nimiösivun toteuttamista varten. Se myös aloittaa sivunumeroinnin alusta eli nolaa `page`-laskurin arvon. Ympäristö `abstract` toimii `article`- ja `report`-dokumenttiluokissa ja on tarkoitettu tieteellisen artikkelin tiivistelmän kirjoittamiseen. Molempien edellä mainittujen ympäristöjen käyttö on vapaaehtoista. Nimiösivun ja artikkelin tiivistelmän voi aivan hyvin toteuttaa ilman näitäkin.

Teoksen nimiösivu – jos sellainen on – kannattanee lisätä pdf-tiedoston sisäiseen sisällysluetteloon, jotta lukija voi helposti palata alkuun myös luettelon kautta. Normaaliin ladottavaan sisällysluetteloon tuskin koskaan merkitään nimiösivua. Pdf:n sisällysluetteloon saa oman merkinnän esimerkiksi seuraavalla komennolla:

```
\pdfbookmark[0]{Nimiö}{sivu/nimiö}
```

Edellä mainittu komento `\pdfbookmark` täytyy sijoittaa Latex-lähdedokumentissa nimenomaan nimiösivulle, koska sisällysluettelomerkintä on samalla linkki kyseiseen kohtaan dokumentissa. Komento kuuluu `hyperref`-pakettiin, ja sitä käsitellään tarkemmin pdf:n asetusten yhteydessä luvussa 3.3.

4.4.2 Otsikointi

Otsikoiden tehtävänä on tietenkin jakaa sisältö mielekkäisiin osiin ja ilmaista tekstikokonaisuuksien aihe. Ne auttavat lukijaa ennakoimaan,

¹⁵ <https://www.ctan.org/pkg/titling>

Komento	Taso (article)	Taso (book, report)	Merkitys
<code>\part</code>	0	-1	teoksen osa
<code>\chapter</code>		0	pääluku
<code>\section</code>	1	1	otsikko 1, luku
<code>\subsection</code>	2	2	otsikko 2, alaluku
<code>\subsubsection</code>	3	3	otsikko 3, alaluvun alaluku
<code>\paragraph</code>	4	4	kappaleotsikko 1
<code>\subparagraph</code>	5	5	kappaleotsikko 2

Taulukko 4.5: Otsikointikomennot ja niitä vastaavat tasot Latexin otsikkohierarkiassa. Dokumenttiluokassa `article` ei ole päälukuja (`\chapter`).

mitä on tulossa. Otsikot havainnollistavat myös asioiden hierarkkisuu-
ta: ylemmätasoiset otsikot nimeävät yläkäsitteitä tai laajempia koko-
naisuuksia, alemmätasoiset otsikot alakäsitteitä ja pienempiä koko-
naisuuksia.

Jos kirjoittaa Latexilla vähänkään otsikoita, kannattaa ladata mukaan paketti `hyperref`. Se huolehtii, että Latexiin merkityt otsikot tuodaan mukaan pdf-tiedoston sisäiseen sisällysluetteloon ja että dokumenttiin mahdollisesti ladottavassa sisällysluettelossa otsikot ovat automaattisesti myös linkkejä, joilla voi hypätä kyseiseen lukuun. Pdf-tiedoston selailu on näin mukavampaa. `hyperref`-pakettia ja pdf-tiedoston ominaisuuksia käsitellään tarkemmin luvussa 3.3.

Latexissa eri otsikoiden tasoille on omat komentonsa, jotka on koottu taulukkoon 4.5 – suurimmasta otsikosta pienimpään. Ylin otsikkotasoa on teoksen osa, ja sitä merkitään komennolla `\part`. Tämän komennon käyttö on vapaaehtoista, mutta sillä voi osoittaa laajan teoksen osia. Osat numeroidaan oletuksena roomalaisilla järjestyslukuilla: ”Osa I”, ”Osa II”, ”Osa III” jne. Ne eivät vaikuta alemmätasoisien otsikoiden numerointiin, eli osan vaihtuminen ei oletusasetuksilla nollaa alemmätasoisien lukujen laskuria.

Otsikkotasoa `\chapter` tarkoittaa kirjan päälukua, joka alkaa oletuksena puhtaalta sivulta. Otsikkoa ennen suoritetaan automaattisesti sivunvaihtokomento `\clearpage` tai `\cleardoublepage` (luku 4.3). `\chapter`-tasoiset otsikot saavat oletuksena numeroinnin ”Luku 1”, ”Luku 2”,

”Luku 3” jne. Dokumenttiluokassa `article` ei tätä otsikkotasoa ole lainkaan.

Seuraavat otsikkotasot `\section`, `\subsection` ja `\subsubsection` ovat tavallisimpia tekstin väliotsikoita. Nekin saavat oletuksena numeroinnin, jossa pisteellä erotetaan mahdolliset alaluvut: 1.1, 1.2, 1.2.1, 1.2.2, 1.3 yms.

Otsikoinnissa on tärkeää säilyttää suora hierarkkinen suhde otsikkotasojen välillä: alemmantasoista otsikkoa ei saa käyttää ennen kuin on käytetty yhtä astetta ylempää otsikkotasoa. Täytyy siis olla ensin `\section` ennen kuin voi olla `\subsection`-komentoja. Muuten otsikoiden numerointi ei toimi oikein. Poikkeus on `\part`, joka ei vaikuta numerointiin ja jonka käyttö on muutenkin vapaaehtoista.

Otsikkotasot `\paragraph` ja `\subparagraph` eivät ole ihan perinteisiä otsikoita, vaan ne ladotaan tekstikappaleen alkuun. Nekin kyllä erotuvat leipätekstistä ja voivat toimia otsikon kaltaisessa tehtävässä. Oletuksena kappaleotsikoita ei numeroida eivätkä ne tule mukaan sisällysluetteloon. Oletusasetuksia on kuitenkin mahdollista muuttaa, ja tarvittaessa `\paragraph` ja `\subparagraph` voivat toimia normaaleina otsikoina tason `\subsubsection` jälkeen. Kovin syvää otsikkorakennetta ei yleensä suositeta, koska lukijan voi olla vaikeaa hahmottaa monitasoisia kokonaisuuksia.

Otsikkokomennoille annetaan tyypillisesti vain yksi argumentti, joka on kyseisen otsikon teksti. Otsikkoon ei ole tarkoitus kirjoittaa kovin monimutkaisia fontti- eikä muotoilukomentoja, vaan mieluiten ainoastaan otsikon teksti. Typografinen muotoilu tehdään muilla keinoilla, joita esitellään luvussa 4.4.4. Yksinkertaiset korostuskomennot ja tavutusvihjeet (`\-`) kuitenkin toimivat yleensä. Otsikkokomennon argumentissa olevien hauraiden komentojen (luku 2.2.3) eteen täytyy kirjoittaa komento `\protect`.

```
\section{Tämä teksti on otsikko}
```

Otsikko tulee automaattisesti mukaan teoksen sisällysluetteloon, jos sellainen on olemassa (luku 4.5). Otsikko näkyy myös pdf-tiedoston sisällysluettelossa, jos `hyperref`-paketti on käytössä (luku 3.3). Lisäksi otsikon tekstiä voi käyttää ristiviittauksissa (luku 4.10).

Otsikkokomennoille voi antaa yhden valinnaisen argumentin, jolla ilmaistaan otsikosta lyhempi tai yksinkertaisempi versio. Tällöin sisällysluettelossa ja ristiviitteissä näytetään lyhempi otsikko ja tekstiin ladotaan varsinainen otsikko. Tätä ominaisuutta on syytä hyödyntää myös silloin, kun varsinainen otsikko sisältää otsikkotekstin lisäksi muotoilukomentoja. Silloin valinnaiseen argumenttiin kirjoitetaan vain puhdasta tekstiä sisältävä versio otsikosta.

```
\section[Lyhyt otsikko]{Tämä tässä on pitkä otsikko}
```

Kaikille otsikkokomennoille on olemassa myös tähtiversio eli komentovaihtoehto, jonka nimen lopussa on tähti: `\part*`, `\chapter*`, `\section*`, `\subsection*`, `\subsubsection*`, `\paragraph*`, `\subparagraph*`. Nämä vaihtoehdot latovat otsikon, jolla ei ole numerointia ja jota ei lisätä sisällysluettelo.

Pelkän otsikon numeroinnin poistamiseen ei ehkä kannata käyttää näitä tähtiversioita, koska yleensä otsikko kuitenkin halutaan sisällysluettelo. Otsikoiden numeroinnin saa pois keinoilla, jotka neuvotaan luvussa 4.4.3. Jos kuitenkin haluaa käyttää tähdellistä otsikkokomentoa ja haluaa otsikon myös sisällysluettelo, täytyy se tehdä seuraavalla tavalla:

```
\section*{Tässäpä numeroimaton otsikko}  
\phantomsection  
\addcontentsline{toc}{section}{Tässäpä numeroimaton otsikko}
```

Edellä olevan esimerkin komento `\addcontentsline` lisää merkinnän sisällysluettelo. Komennon ensimmäinen argumentti on `toc` (table of contents), joka on sisällysluetteloiden tekemisessä tarvittavan väliaikaistiedoston päte. Esimerkissä lisättävä sisällysluettelon merkintä on tasoltaan `section` eli vastaa muita `\section`-komennolla syntyviä sisällysluettelon kohtia. Komento `\phantomsection` kuuluu `hyperref`-pakettiin, ja se lisää pdf-tiedostoon näkymättömän ankkurin tähän kohtaan dokumentissa. Sitä tarvitaan, jotta sisällysluettelon merkintä viittaisi oikealle sivulle ja toimisi myös linkkinä, joka tuo juuri tähän kohtaan.

4.4.3 Otsikoiden numerointi

Oletusasetuksilla Latex numeroi otsikot (`\chapter`, `\section` jne.) automaattisesti, mutta kirjoittaja voi vaikuttaa numerointiin monella tavalla, kuten siihen, mihin otsikkotasoon saakka numerointi ylittää ja millä tyyllillä otsikoiden numerot ylipäätään ladotaan.

Laskuri `secnumdepth` on tarkoitettu kirjoittajan asetettavaksi, ja sillä määritetään, mihin otsikkotasoon saakka otsikot numeroidaan. Tasonumerot ovat taulukossa 4.5 (s. 145). Seuraava esimerkki määrittelee, että vain tasoon 1 (`\section`) saakka otsikot numeroidaan. Alemmanta-
soiset otsikot (`\subsection` ym.) eivät saa numeroa.

```
\setcounter{secnumdepth}{1}
```

Asetusta voi muuttaa kesken dokumentin. Muuttaminen voi olla mielekästä esimerkiksi laajan teoksen lopussa, jossa on mahdollisesti lähdeluettelo, asiahakemistoja tai liitteitä, joille ei ehkä haluta samanlaista otsikkonumerointia kuin varsinaisille sisältöluvuille.

Latexin otsikoiden numerointi noudattaa yleistä tietokirjallisuuden käytäntöä, eli numeroinnissa käytetään arabialaisia numeroita ja eri tasoja ilmaisevat numerot erotetaan toisistaan pisteellä: 1.1, 1.2, 1.2.1, 1.2.2 jne. Sisäisesti Latex käyttää laskureita, joilla on sama nimi kuin vastaavalla otsikointikomennolla: `part`, `chapter`, `section`, `subsection`, `subsubsection`, `paragraph`, `subparagraph`.

Laskureiden toimintaan voi perehtyä tarkemmin luvun 2.5 avulla, mutta kerrataan tässä yhteydessä yksi asia. Laskurin arvon voi latioa dokumenttiin käyttämällä komentoa, joka alkaa kirjaimilla `\the` ja jatkuu laskurin nimellä. Niin otsikkokomennot juuri tekevät: esimerkiksi `\section`-otsikon kohdalla suoritetaan automaattisesti `\thesection`, joka tuottaa otsikon numeron. Vastaavasti `\subsection`-komento latioo numeron käyttämällä komentoa `\thesubsection` jne.

Kirjoittaja voi halutessaan määritellä nämä laskurin latomiskomennot uudelleen. Jos esimerkiksi teoksen osat (`\part`) halutaan ilmaista arabialaisin numeroin eikä roomalaisin (kuten on oletus), tehdään se seuraavan esimerkin mukaisesti. Esimerkissä käytettävä komento `\arabic` latioo laskurin arvon arabialaisilla numeroilla.

```
\renewcommand{\thepart}{\arabic{part}}
```

Joskus ehkä halutaan, että otsikoissa eri tasojen numeroita ei eroteta toisistaan pisteellä (2.1) vaan yhdysmerkillä tai ajatusviivalla (2–1). Tällainen asetus saadaan määrittelemällä alemmantasoisien laskurien latomiskomennot uudelleen, esimerkiksi seuraavasti:

```
\renewcommand{\thesubsection}{\thesection--\arabic{subsection}}
\renewcommand{\thesubsubsection}{%
  \thesubsection--\arabic{subsubsection}}
```

Otsikkonumeroinnin poistamista ei pidä toteuttaa siten, että määrittelee edellä mainitut `\the`-alkuiset komennot tyhjäksi. Numerointi poistetaan laskurin `secnumdepth` avulla määrittämällä sen arvo tarpeeksi pieneksi. Sekin on syytä muistaa, että nämä `\the`-alkuiset komennot on tarkoitettu vain laskurien arvon latomiseen. Komennon määritelmään ei pidä kirjoittaa typografiseen muotoiluun liittyviä komentoja. Otsikoiden ulkoasuun vaikutetaan muilla keinoilla.

4.4.4 Otsikoiden ulkoasu

Otsikoiden typografiaan eli esimerkiksi kirjainperheen ja -leikkauksen valintaan ja välistyksiin on kätevintä käyttää `titlesec`¹⁶-pakettia. Perus Latexissakin on kyllä otsikoiden muotoiluun tarkoitettu komento `\@startsection`, mutta sen käyttö on hankalampaa eikä se toimi kaikilla otsikkotasolla. Tässä oppaassa keskitytään `titlesec`-pakettiin.

Otsikoiden perusmuotoilu on helpointa toteuttaa `\titleformat*`-komennon avulla. Sille annetaan argumenteiksi otsikkokomento ja sitä vastaavat muotoilukomennot. Seuraava esimerkki vaihtaa otsikoihin groteskin kirjainperheen (`\sffamily`) ja hieman toisistaan poikkeavan kirjainleikkauksen.

```
\titleformat*{\section}      {\sffamily\bfseries\Large}
\titleformat*{\subsection}  {\sffamily\bfseries\itshape\large}
\titleformat*{\subsubsection}{\sffamily\bfseries\normalsize}
```

Monipuolisemmat muotoiluasetukset saa toteutettua komennon tähdettömällä versiolla `\titleformat`. Tämän komentoversion monista argumenteista ja ominaisuuksista kannattaa lukea `titlesec`-paketin ohjekirjasta, mutta esimerkiksi 4.5 selvinnevät perustoiminnot.

¹⁶ <https://www.ctan.org/pkg/titlesec>

1	<code>\titleformat{\section}</code>	% otsikkotaso: \section
2	<code>[hang]</code>	% muoto: riippuva sisennys
3	<code>{\sffamily\bfseries\Large</code>	% ulkoasuomerkinnät
4	<code>\raggedright}</code>	
5	<code>{\thesection}</code>	% numerointi
6	<code>{.8em}</code>	% numeroinnin ja otsikkotekstin väli (mitta)
7	<code>{}</code>	% koodia ennen otsikkotekstiä
8	<code>[]</code>	% koodia otsikon jälkeen

Esimerkki 4.5: Otsikoiden ulkoasuun voi vaikuttaa monipuolisesti `\titleformat`-komennolla, joka kuuluu `titlesec`-pakettiin.

Luvun numeroinnin (1.1, 1.2 jne.) tyyliin voi vaikuttaa komennolla `\titlelabel`. Komento suoritetaan yleensä vain kerran, ja se vaikuttaa kaikentasoisiiin otsikoihin. Tätä komentoa ei tarvita, jos käyttää edellä mainittua ”tähdetöntä” `\titleformat`-komentoa, koska se sisältää jo toiminnot myös lukujen numeroinnin asetuksiin.

```
\titlelabel{\thetitle\hspace{1em}}
```

Edellä olevan komennon argumentissa käytetty komento `\thetitle` on tarkoitettu vain tämän komennon ja `titlesec`-paketin sisäiseen käyttöön. Se lataa kyseisen luvun numeron, olipa otsikkotaso mikä hyvänsä. Sen perään esimerkissä asetetaan 1 em:n suuruinen väli `\hspace`-komennolla.

Otsikoiden yhteydessä ladottavaan sisennykseen ja pystysuuntaiseen väliin vaikutetaan komennolla `\titlespacing*`. Sille annetaan argumenteiksi otsikkokomento ja halutut mitat järjestyksessä vasen sisennys, yläpuolinen väli ja alapuolinen väli.

```
\titlespacing*{\section}{0mm}{3ex plus 2bp minus 1bp}{2ex plus 1bp}
```

Pystysuuntaiset mitat on yleensä hyvä asettaa venyviksi (luku 2.4.3), jotta Tex saa pelivaraa sivujen latomiseen. Otsikon yläpuolella pitäisi olla isompi väli kuin alapuolella, koska otsikko kuuluu kiinteämmin sitä seuraavaan tekstiin. Pystysuuntaisten välien määrittämisessä voi käyttää *-lyhennettä, joka tarkoittaa oletuksena ex-mittayksikköä ja sallii hieman venymistä.

```
\titlespacing*{\section}{0mm}{*3}{*2}
```

Tähdellinen `\titlespacing*`-komento poistaa otsikon jälkeen tulevasta tekstikappaleesta ensimmäisen rivin sisennyksen (luku 4.1.3). Tähdetön komentoversio `\titlespacing` sen sijaan säilyttää kappaleen sisennyksen. Sitä ei liene tarvetta käyttää koskaan, sillä otsikoiden jälkeen tulevaa tekstikappaletta ei kuulu sientää.

Teoksen osien (`\part`) vaihtuessa sivulle ladotaan oletuksena teksti kuten ”Osa I”, ”Osa II” jne. Mainittu sana ”Osa” tulee suomen kieliasetuksista ja komennosta `\partname`. Kaikkien kielten asetukset nimittäin määrittelevät komennon `\partname` siten, että se sisältää ja latoo kyseiseen kieleen sopivan ilmauksen. Vastaavasti päälukujen (`\chapter`) yhteydessä ladotaan suomen kieliasetuksilla ilmaus ”Luku 1”, ”Luku 2” jne. Sana ”Luku” tulee komennosta `\chaptername`.

Jos omassa dokumentissa haluaa kutsua teoksen osia tai päälukuja joksikin muuksi, voi määritellä edellä mainitut komennot uudelleen. Seuraavassa esimerkissä voisi olla jokin romaani, jossa teoksen osat ovat ”kausia” ja pääluvut ovat ”päiviä”.

```
\addto{\captionsfinnish}{  
  \renewcommand{\partname}{Kausi}  
  \renewcommand{\chaptername}{Päivä}  
}
```

Edellisessä esimerkissä `\renewcommand`-komennot sijoitettiin `\addto`-komennon toiseen argumenttiin. Komento `\addto` täytyy suorittaa lähdedokumentin esittelyosassa, ja tässä esimerkissä se lisää uusia komentoja suomen kielen asetuksiin (`\captionsfinnish`), niin että ne tulevat voimaan samalla kun suomen kieliasetuksetkin. `\addto` kuuluu `polyglossia`- ja `babel`-paketteihin, joita käsitellään tarkemmin kieliasetusten yhteydessä luvussa 3.5.

Otsikoiden yhteyteen saa automaattisen sivunvaihdon määrittämällä komennon, jonka nimen alussa on otsikkokomennon nimi ja lopussa sana `break`. Tavallisin lienee `\sectionbreak`. Komennon määritelmään kirjoitetaan jokin sivuvaihtokomento kuten `\clearpage`. Sivunvaihtokomentoja käsitellään tarkemmin luvussa 4.3.

```
\newcommand{\sectionbreak}{\clearpage}
```

4.4.5 Esittely, pääluvut, liitteet ja luettelot

Dokumenttiluokassa `book` (luku 3.1.1) on käytettävissä kolme lisäkomentoa dokumentin erilaisten osien ilmaisemiseen: `\frontmatter`, `\mainmatter` ja `\backmatter`. Niillä jaetaan laaja teos esittelysivuihin, varsinaisiin sisältölokuihin ja loppuosaan kuten liitesivuihin ja erilaisiin luetteloihin. Lisäksi muissakin dokumenttiluokissa on komento `\appendix`, jolla voi aloittaa liitesivuja sisältävän osan dokumentissa. Näiden komentojen käyttö on vapaaehtoista. Luvussa 4.4.6 käsitellään tyypillisen suomalaisen tietokirjan rakennetta ja teknistä toteuttamista.

Dokumentin alkuun voi kirjoittaa komennon `\frontmatter`, joka asettaa sivunumeroiden tyyliksi roomalaiset numerot (i, ii, iii jne.) ja poistaa pääluvuilta (`\chapter`) numeroinnin. Tässä osassa ovat teoksen nimiö ja perustiedot, sisällysluettelo, tietokirjan tai tutkimuksen tiivistelmä ja mahdollisesti teoksen esipuhe tai muu vastaava esittelyteksti, joka ei ole vielä varsinaista sisältöä.

Jos dokumentissa käytetään `\frontmatter`-komentoa, täytyynee varsinaisten sisältölokujen alkuun kirjoittaa komento `\mainmatter`, joka nolaa sivunumerolaskurin, asettaa numeroinnin tyyliksi arabialaiset numerot (1, 2, 3 jne.) ja kytkee päälle lukujen numeroinnin. Sen jälkeen ensimmäinen `\chapter`-komennolla tehty luku on siis ”Luku 1”.

Sisältösivujen lopussa voi olla komento `\appendix`. Se ensinnäkin nolaa `book`- ja `report`-luokissa päälukujen laskurin `chapter`. Sen sijaan `article`-luokassa se nolaa `section`-laskurin. Sen lisäksi `\appendix`-komento muuttaa lukujen numerointityyliksi kirjaimet (A, B, C jne.) ja päälukujen nimeksi ”Liite”, joka tulee komennosta `\appendixname`. Näin teoksen lopussa `\appendix`-komennon jälkeen pääluvut tulevat nimetyksi uudella tavalla: ”Liite A”, ”Liite B” jne.

Sisältösivujen ja mahdollisten liitteiden jälkeen voi olla hyödyllistä käyttää komentoa `\backmatter`, joka lopettaa päälukujen numeroinnin kokonaan. Tähän osaan voisi sijoittaa ainakin asiahakemistot sekä kirjallisuus- tai lähdeluettelon.

4.4.6 Tyypillinen tietokirjan rakenne

Latexin komennot `\frontmatter` ja `\mainmatter` (luku 4.4.5) eivät ihan vastaa suomalaista tietokirjojen käytäntöä. Suomessa teoksen alun si-

vunumeroita ei ole tapana ilmaista roomalaisilla numeroilla eikä sivunumerointia nollata sisältöluokien alkaessa. Esimerkkiin 4.6 on koottu varsin tyypillinen suomalainen tietokirjojen rakenne. Esimerkissä ei käytetä lainkaan komentoja `\frontmatter`, `\mainmatter` eikä `\backmatter`, vaan lukujen numerointiin vaikutetaan suoraan asettamalla laskuriin `secnumdepth` sopiva arvo. Tätä laskuria käsitellään otsikoiden yhteydessä luvussa 4.4.3.

Suomalaisissa kirjoissa sivunumeroinnin laskuri alkaa ensimmäiseltä sivulta, johon on painettu jotakin, yleensä pelkkä teoksen nimi tai kirjailijan nimi. Sivun 1 ei ole vielä kirjan kansi, koska kirjapainossa kannet lisätään ikään kuin tekijöiden työn ympärille. Aikakauslehdissä kansi on kuitenkin sivu 1.

Sivunumerot tulevat näkyviin hieman vaihtelevasti: joskus sivunumerot näytetään sisällysluettelon alusta saakka; joskus ne näytetään vasta sisällysluettelon jälkeen esipuheessa; joskus esipuhe on ennen sisällysluetteloja ja sivunumerot näkyvät esipuheen alusta saakka. Esimerkissä 4.6 sivutyylillä `plain` kytketään päälle sisällysluettelon alkaessa, eli sivunumerotkin alkavat sen myötä näkyä. Sivutyylejä ja muita ylä- ja alatunnisteisiin liittyviä asioita käsitellään tarkemmin luvussa 3.2.4.

Mikäli haluaa, että sisällysluettelossa ei vielä näytetä sivunumeroita, joutuu dokumenttiluokissa `book` ja `report` käyttämään vähän epätavallista keinoa halutun lopputuloksen saavuttamiseksi. Sellainen neuvotaan sisällysluettelon käsittelyn yhteydessä esimerkissä 4.7 (s. 156).

4.5 Sisällysluettelo

Dokumentin sisällysluettelo syntyy helposti yhdellä komennolla: `\tableofcontents`. Otsikkokomennoilla (luku 4.4.2) tehdyt otsikot tulevat automaattisesti sisällysluetteloon. Jos käytössä on paketti `hyperref` (luku 3.3), sisällysluettelon otsikot ovat myös linkkejä, jotka vievät kyseiseen kohtaan dokumentissa. Oletuksena syntyvä sisällysluettelo on tyyliltään neutraali, ja se sopinee sellaisenaan useimpiin töihin. Ulkoasua voi kyllä muokatakin varsin helposti.

```
% Ladotaan sisällysluettelo tähän kohtaan.  
\tableofcontents
```

```

1 \documentclass{book}
2
3 \begin{document}
4
5 \pagestyle{empty} % sivunumerot pois näkyvistä
6 % nimiösivu yms.
7
8 \cleardoublepage
9 \setcounter{tocdepth}{2} % sisällysluettelon syvyys
10 \pagestyle{plain} % sivunumerot näkyviin
11 \tableofcontents % sisällysluettelo
12
13 \setcounter{secnumdepth}{-1} % lukujen numerointi pois
14
15 \chapter{Esipuhe}
16 % esipuheen teksti
17
18 \setcounter{secnumdepth}{2} % aloitetaan lukujen numerointi
19
20 \chapter{Ensimmäinen pääluke}
21 % ...
22 \chapter{Toinen pääluke}
23 % ...
24
25 % mahdollisesti liitteet
26 \appendix
27 \chapter{Tärkeä liite}
28 % ...
29 \chapter{Tosi tärkeä liite}
30 % ...
31
32 % Kirjallisuusluettelot, asiahakemistot yms.
33 \setcounter{secnumdepth}{-1} % lukujen numerointi pois
34
35 \chapter{Kirjallisuutta}
36 \printbibliography
37
38 \chapter{Asiahakemisto}
39 \printindex
40
41 \end{document}

```

Esimerkki 4.6: Tyypillinen suomalainen tietokirjojen rakenne sekä sivujen ja lukujen numerointikäytännöt.

Laskurin `tocdepth` avulla kirjoittaja voi määritellä, mihin tasoon saakka sisällysluettelossa näytetään otsikoita. Esimerkiksi jonkin kuvitteellisen romaanin (dokumenttiluokka `book`) sisällysluettelossa voisi näyttää ainoastaan osat (`\part`) ja pääluvut (`\chapter`) asettamalla laskurin arvoksi nolla. Arvo nolla viittaa nimenomaan `\chapter`-tasoiisiin otsikoihin. Otsikoiden tasonumerot voi tarkistaa taulukosta 4.5 (s. 145).

```
\setcounter{tocdepth}{0}
```

Sisällysluettelolle ladotaan automaattisesti otsikko, joka tulee komennosta `\contentsname`. Komento on määritelty kieliasetuksissa, ja se latoo suomen kielessä sanan ”Sisältö” tai ”Sisällys”.¹⁷ Jos otsikkoon ei ole tyytyväinen, voi lähdedokumentin esittelyosassa `\addto`-komennon avulla muuttaa kieliasetuksia. Seuraavassa esimerkissä muutetaan sisällysluettelon otsikko suomen kielen osalta:

```
\addto{\captionsfinnish}{  
  \renewcommand{\contentsname}{Sisällysluettelo}  
}
```

Dokumenttiluokissa `book` ja `report` (luku 3.1.1) on sellainen ominaisuus tai kummallisuus, että sisällysluettelon alkusivulla suoritetaan automaattisesti komento `\thispagestyle{plain}`. Tämän vuoksi sisällysluettelon ensimmäinen sivu on aina `plain`-tyylinen eli alatunnisteessa näkyy sivunumero, vaikka ennen sisällysluetteloä olisikin asettanut sivutyyliksi jonkin muun. Monisivuisen sisällysluettelon seuraavilla sivuilla sen sijaan noudatetaan aiemmin `\pagestyle`-komennolla asetettua sivutyylä.

Jotta edellä mainitusta kummallisuudesta pääsisi eroon, täytyy määritellä väliaikaisesti `\thispagestyle`-komento toisenlaiseksi. Väliaikainen komennon määrittely saadaan aikaan aaltosulkeilla (luku 2.1.6), joiden sisään komennon uudelleen määrittely ja sisällysluettelon latomiskomento sijoitetaan. Esimerkistä 4.7 selviää, miten se tehdään. Tässä esimerkissä komennon määritelmä jätettiin tyhjäksi, koska sen ei ha-

¹⁷ Se, kumpi otsikko on käytössä, riippuu kielipakettien `polyglossia` ja `babel` versiosta. Vuonna 2020 kumpaankin kielipakettiin on muutettu otsikoksi ”Sisällys”, joka on selvästi yleisin sisällysluettelon otsikko.

```

1 \pagestyle{empty}
2 {
3   \renewcommand{\thispagestyle}[1]{} % tyhjä määritelmä
4   \tableofcontents
5 }

```

Esimerkki 4.7: Sisällysluettelon sivutyylin muuttaminen kokonaan empty-tyyliseksi. Komento `\thispagestyle` pitää määritellä väliaikaisesti uudestaan dokumenttiluokissa `report` ja `book`.

luta tekevän mitään. Näin ollen aiemmin annettu `\pagestyle`-komento vaikuttaa koko sisällysluetteloon.

Normaalit otsikot tulevat mukaan sisällysluetteloon itsestään, mutta joskus voi joutua lisäämään luetteloon kohtia myös keinotekoisesti. Se tehdään komennolla `\addcontentsline`, jonka käytöstä on esimerkki luvussa 4.4.2. Pelkkään pdf-tiedoston sisäiseen sisällysluetteloon voi lisätä kohtia komennolla `\pdfbookmark`, joka kuuluu `hyperref`-pakettiin. Tätä komentoa käsitellään luvussa 3.3.

Sisällysluettelon ulkoasun muokkaamista varten on paketti `titletoc`¹⁸, jonka tärkeimmän komennon (`\titlecontents`) käyttöä esitellään esimerkissä 4.8. Komennolla voi muokata paitsi sisällysluettelon kohtia mutta myös leijuvien osien (luku 4.9) kuten kuvien ja taulukoiden luettelaita.

Esimerkistä selviää, miten `\titlecontents`-komennolla määritellään luettelokohdan vasemman sisennyksen mitta, yläpuolinen ja alapuolinen väli (`\addvspace`, luku 4.1.2), mahdolliset kirjainperheeseen ja -leikkaukseen vaikuttavat komennot kuten `\rmfamily`, `\bfseries` ja `\large` (luku 3.4.3).

`\titlecontents`-komennon argumenteissa käytetään jonkin verran muita komentoja, jotka on tarkoitettu nimenomaan tämän komennon argumentteihin. Komento `\contentslabel` latoo kyseisen otsikon numeron (1.2, 1.3 tms.) tietynlevyiseen tilaan ja sisennystason vasemmalle puolelle (riippuva sisennys). Komento `\contentspage` latoo sivunume-

¹⁸ <https://www.ctan.org/pkg/titletoc>

```

1 \titlecontents{chapter} % otsikon taso: chapter
2 [8mm] % vasen sisennys (mitta)
3 {\addvspace{1.5ex}\rmfamily\bfseries\large} % yläpuolinen koodi
4 {\contentslabel{8mm}} % numeroitu kohta
5 {\hspace{-8mm}} % numeroimaton kohta
6 {\small\titlerule[0bp]\contentspage} % pisteiviiva (pois) ja sivunumero
7 [\addvspace{.5ex}] % alapuolinen koodi
8
9 \titlecontents{section} % otsikon taso: section
10 [8mm] % vasen sisennys (mitta)
11 {\addvspace{.5ex}\rmfamily\normalsize} % yläpuolinen koodi
12 {\contentslabel{8mm}} % numeroitu kohta
13 {} % numeroimaton kohta
14 {\small\titlerule*[3mm]{.}\contentspage} % pisteiviiva ja sivunumero
15 [\addvspace{.2ex}] % alapuolinen koodi
16
17 \titlecontents{subsection} % otsikon taso: subsection
18 [18mm] % vasen sisennys (mitta)
19 {\rmfamily\small} % yläpuolinen koodi
20 {\contentslabel{10mm}} % numeroitu kohta
21 {} % numeroimaton kohta
22 {\small\titlerule*[3mm]{.}\contentspage} % pisteiviiva ja sivunumero
23 [] % alapuolinen koodi

```

Esimerkki 4.8: Sisällysluettelon ulkoasua muokataan `\titlecontents`-komennolla, joka on peräisin `titletoc`-paketista.

ron. Komennolla `\titlerule` voi tehdä katsetta ohjaavan viivan tai pisteiviivan otsikkotekstin ja sivunumeron väliin.

Paketin `titletoc` ohjekirjassa on paljon muitakin toimintoja ja esimerkkejä sisällysluetteloiden ehostamiseen. Ohjeisiin kannattaa tutustua, mikäli kaipaa erikoisempia sisällysluettelomalleja.

4.6 Luettelmat

Latexissa on helppokäyttöiset ympäristöt kolmelle erilaiselle luettelman perustyyppille: numeroimaton (`itemize`), numeroitu (`enumerate`) ja määritelmäluetelma (`description`). Ne riittävät perustarpeisiin ja ovat hieman myös muokattavissa typografisesti. Näitä luettelmien perusympäristöjä käsitellään luvussa 4.6.1.

Perusluetelmien lisäksi on olemassa yleinen luettelujen rakenteluun tarkoitettu ympäristö `list`, jolla voi toteuttaa suunnilleen mitä tahansa luotelmiin tai tekstikappaleisiin liittyviä rakenteita. Teknisesti `list`-ympäristö sisältää kaikki tarvittavat luotelmiin liittyvät ominaisuudet, joten kirjoittaja voi aivan hyvin käyttää pelkästään sitä. Ympäristöä käsitellään luvussa 4.6.2.

4.6.1 Perusympäristöt

Ympäristö `itemize` tekee numeroimattoman luettelun, jonka jokainen kohta merkitään samanlaisella merkillä, esimerkiksi luettelmaympyrällä (`•`). `enumerate` puolestaan on numeroitu luettelo, eli luettelun kohdat saavat järjestysnumeron tai muun järjestystä ilmaisevan merkinnän kuten kirjaimen tai roomalaisen numeron. Ympäristö `description` luo määritelmäluettelun, jossa luotellaan käsitteitä ja niiden määritelmiä. Tämä on käytännössä sama kuin tekstikappaleen riippuva sisennys (luku 4.1.4), jossa kappale alkaa määriteltävällä ilmauksella.

Lähdetiedostossa kaikkien luettelmatyyppien rakenne on samanlainen: luettelmaympäristön sisällä käytetään `\item`-komentoa, joka aloittaa uuden luettelmakohdan. Komennolle ei tarvitse antaa argumenttia, vaan komennon jälkeinen teksti muodostaa luettelun kohdan. Seuraavassa on esimerkki `itemize`-ympäristöstä, mutta `enumerate`-ympäristöäkin käytetään samalla tavalla:

```
\begin{itemize}
\item Luettelun ensimmäinen kohta.
\item Toinen kohta heti perään.
\item Kolmaskin kohta on tarpeen.
\end{itemize}
```

`\item`-komennolle voi antaa hakasulkeissa valinnaisen argumentin. Silloin ei ladota tavanomaista luettelamerkkiä vaan argumentin sisältämä teksti.

```
\item[--] Tässä kohdassa onkin ajatusviiva.
```

Komennon valinnaista argumenttia hyödynnetään varsinkin `description`-ympäristössä, jossa luotellaan sanoja tai käsitteitä ja niiden määritelmiä. Esimerkistä 4.9 selviää, kuinka määriteltävät käsitteet ilmaistaan `\item`-komennon valinnaisessa argumentissa.

```

1 \begin{description}
2 \item[TeX] Tekstidokumenttien ladontaan erikoistunut ohjelmointikieli.
3 \item[Latex] Dokumenttien kirjoittajille tarkoitettu merkintäkieli.
4 \item[Lualatex] Latex-muotoisten lähdedokumenttien kääntäjäohjelma.
5   Sisältää Lua-nimisen ohjelmointikielen.
6 \end{description}

```

⇒ **Tex** Tekstidokumenttien ladontaan erikoistunut ohjelmointikieli.

Latex Dokumenttien kirjoittajille tarkoitettu merkintäkieli.

Lualatex Latex-muotoisten lähdedokumenttien kääntäjäohjelma. Sisältää Lua-nimisen ohjelmointikielen.

Esimerkki 4.9: Käsitteiden määritelmiä ja sen kaltaisia luettelma voi toteuttaa `description`-ympäristön avulla. Käsitteet kirjoitetaan `\item`-komennon valinnaiseen argumenttiin.

Edellä mainittuja luettelmaympäristöjä voi kirjoittaa sisäkkäin: yksi luotelman kohta (`\item`) voi siis aloittaa uuden luettelmaympäristön. Latex osaa käsitellä neljä sisäkkäistä luettelmaa.

Latex sisentää sisäkkäiset luettelmat automaattisesti loogisella tavalla ja vaihtaa eri tasoilla luettelmaehtoien merkkiä tai numerointitapaa. Esimerkiksi uloimmainen `itemize`-ympäristö käyttää luettelmaympyrää (•), mutta sen sisällä oleva käyttää ajatusviivaa (–). Uloimmainen `enumerate`-ympäristö käyttää arabialaisia numeroita (1, 2, 3 jne.) mutta sen sisällä oleva kirjaimia (a, b, c jne.).

Luettelmamerkki ja luettelmien numerointitavat ovat kirjoittajan muuttavissa. Ympäristössä `itemize` luettelmamerkki tulevat komennosta, joka alkaa sanoilla `\labelitem` ja jatkuu luotelman tasoa ilmaisevalla roomalaisella numerolla `i–iv`. Komennot voi määrittellä uudelleen haluamallaan tavalla, kuten esimerkki 4.10 osoittaa.

`enumerate`-ympäristössä numeroidut kohdat tulevat komennosta, joka alkaa sanoilla `\labelenum` ja jatkuu luotelman tasoa ilmaisevalla roomalaisella numerolla `i–iv`. Numeroidussa luettelmassa käytetään laskureita `enumi`, `enumii`, `enumiii` ja `enumiv`, joten luettelmaehtoien täytyy laitoa näiden laskurien arvo jossakin muodossa. Esimerkki 4.10 selvittää asiaa.

```

1 % itemize-ympäristön luettelamerkki
2 \renewcommand{\labelitemi} {\textbullet} % uloimmainen luettelma
3 \renewcommand{\labelitemii} {\normalfont\bfseries\textendash}
4 \renewcommand{\labelitemiii} {\textasteriskcentered}
5 \renewcommand{\labelitemiv} {\textperiodcentered}
6
7 % enumerate-ympäristön numerointitapa
8 \renewcommand{\labelenumi} {\arabic{enumi}.} % uloimmainen luettelma
9 \renewcommand{\labelenumii} {\alph{enumii}}
10 \renewcommand{\labelenumiii} {\roman{enumiii}.}
11 \renewcommand{\labelenumiv} {\Alph{enumiv}.}

```

Esimerkki 4.10: Luettelamerkkien ja numerointitapojen muuttaminen `itemize-` ja `enumerate-`ympäristöissä. Esimerkissä näkyvät oletusarvot.

Numeroiduista luettelmakohdista huolehtivat laskurit nolldataan automaattisesti aina ympäristön alussa. Joskus voi kuitenkin olla tarpeen aloittaa luettelmakohdat jostain muusta kuin luvusta 1 tai kirjaimesta *a*. Silloin kirjoittajan täytyy asettaa laskuri haluamaansa arvoon `enumerate-`ympäristön sisällä. Seuraava esimerkki asettaa uloimman luettelman `enumi-`laskurin arvoon 3, joten ensimmäiseksi luettelmakohdaksi tulee 4.

```

\begin{enumerate}
  \setcounter{enumi}{3}
  \item Tämä on luettelmakehta numero 4.
\end{enumerate}

```

Kirjoittaja voi vaikuttaa myös luettelmakehtien sisennykseen ja pystysuuntaisiin väleihin. Sisennykseen vaikuttavat mitat alkavat sanoilla `\leftmargin` ja jatkuvat roomalaisella numerolla *i–iv*, joka ilmaisee luettelman tasoa. Uloimmaisen luettelman sisennys ilmaistaan suhteessa sivun tekstialueen reunaan; sisempien luettelmakehtien sisennys ilmaistaan suhteessa edellisen tason sisennykseen. Oletuksena Latexin sisennykset ovat melko suuria, joten niitä on usein tarpeen pienentää, esimerkiksi seuraavan esimerkin mukaisesti:

```

\setlength{\leftmargini} {1.5em} % uloimmainen luettelma
\setlength{\leftmarginii} {1.5em}
\setlength{\leftmarginiii}{1.5em}
\setlength{\leftmarginiv} {1.5em}

```


Edellä olevat sisennysasetukset toimivat ympäristöissä `itemize`, `enumerate` ja `description`. Mainituissa ympäristöissä voi säätää pystysuuntaisia välejä mittojen `\parsep`, `\itemsep` ja `\parskip` avulla. Mittojen sijainti luettelussa on nähtävissä kuvassa 4.2 (s. 164). Mittojen asettamiskomennot on syytä sijoittaa luettelmaympäristön sisään heti `\begin`-komennon jälkeen.

```
\setlength{\parsep}{0bp} % kappaleiden väli
\setlength{\itemsep}{3bp} % luettelakohtien väli
\setlength{\parskip}{1.2ex} % väli luotelman alussa ja lopussa
```

4.6.2 Omat luettelmat (list)

Ympäristö `list` on yleistyökalu, joka soveltuu moniin asioihin: se on tarkoitettu erilaisten luettelmien rakentamiseen, mutta sen avulla muotoilla mitä tahansa tekstikappaleita. Esimerkiksi luvussa 4.6.1 käsitellyt luettelmat voi toteuttaa myös `list`-ympäristön avulla – samoin luvuissa 4.1.4 ja 4.1.5 käsitellyt riippuvat sisennykset ja lohkolainaukset.

Tyypillisesti `list`-ympäristöä ei käytetä dokumentin kirjoittamisessa sellaisenaan, vaan kirjoittaja määrittelee oman helppokäyttöisen luettelma- tai muun ympäristön, ja `list`-ympäristöä hyödynnetään ympäristön määrittelemisessä. Omien ympäristöjen tekemistä käsitellään yleisesti luvussa 2.3, mutta tämän alaluvun loppupuolella on esimerkki, joka liittyy nimenomaan `list`-ympäristöön.

Seuraavassa on esimerkki `list`-ympäristön rakenteesta. Se muistuttaa luvussa 4.6.1 käsiteltyjä perusluetelmia, mutta lisäksi sille täytyy antaa kaksi argumenttia. Niiden avulla vaikutetaan luotelman asetuksiin.

```
\begin{list}{merkki}{asetukset}
\item ...
\end{list}
```

Edellisen esimerkin argumentti `merkki` sisältää tekstiä tai koodia, joka muodostaa luettelmakohdan merkin. Numeroimattomassa luettelussa siihen voi laittaa esimerkiksi luettelmaympyrän eli komennon `\textbullet` tai ajatusviivan (`--`). Numeroidussa luettelussa siihen kirjoitetaan komento, joka latio jonkin laskurin arvon.

```

1 \begin{list}{\arabic{enumi}}{
2   \usecounter{enumi}           % käytetään enumi-laskuria
3   \setlength{\leftmargin}{1.5em} % vasen sisennys
4   \setlength{\itemsep}{.2ex}   % luettelmakohtien väli
5   \setlength{\parsep}{0ex}    % luettelmakappaleiden väli
6 }
7 \item Luetelman ensimmäinen kohta.
8 \item Tässä on vielä toinen
9 \item ja kolmas kohta.
10 \end{list}

```

- ⇒
- 1) Luetelman ensimmäinen kohta.
 - 2) Tässä on vielä toinen
 - 3) ja kolmas kohta.

Esimerkki 4.11: `list`-ympäristön argumenttien avulla vaikutetaan luettelmakoh-
tien merkintätapaan ja mittoihin.

`list`-ympäristön toinen argumentti asetukset sisältää mitä hyvän-
sä koodia, jolla vaikutetaan ympäristön asetuksiin. Tyypillisesti siinä
määritellään käytettävä luettelmalaskuri sekä asetetaan luotelman ulko-
asuun vaikuttavat mitat sopivaksi. Mahdollisesti siinä myös määritel-
lään luettelmamerkkien latomiseen vaikuttava komento `\makeLabel`.

Esimerkissä 4.11 on yksinkertainen numeroitu luotelma, joka hyö-
dyntää olemassa olevaa `enumi`-laskuria (luku 4.6.1). Laskurin arvo lado-
taan arabialaisilla numeroilla (`\arabic`). Laskurin `enumi` sijasta voi käyt-
tää mitä tahansa omaakin laskuria, kunhan luo sen etukäteen komen-
nolla `\newcounter` (luku 2.5). Esimerkissä asetetaan myös eräitä luotel-
man sisäisiä mittoja, joista on lisätietoa myöhemmin tässä alaluvussa
sekä kuvassa 4.2.

Luettelamerkit `list`-ympäristö luo siten, että se suorittaa aina ko-
mennon `\makeLabel` ja antaa sille argumentiksi kulloisenkin luettelma-
merkin eli `list`-ympäristön ensimmäisenä argumenttina annetun teks-
tin tai `\item`-komennon valinnaisena argumenttina olevan tekstin. Lue-
telmamerkki ladotaan sille varatun alueen oikeaan reunaan. Oletukse-
na `\makeLabel`-komento toimii ikään kuin se olisi määritelty seuraavalla
tavalla:

```
\renewcommand{\makeLabel}[1]{\hfill #1}
```

```

1 \begin{list}{}{
2   \renewcommand{\makelabel}[1]{\textsc{#1:}}
3   \setlength{\leftmargin}{1.5em}
4   \setlength{\labelwidth}{1.5em}
5   \setlength{\itemindent}{1em}
6   \setlength{\labelsep}{1em}
7   \setlength{\itemsep}{.2ex}
8 }
9 \item[TeX] Tekstidokumenttien ladontaan erikoistunut ohjelmointikieli.
10 \item[Latex] Dokumenttien kirjoittajille tarkoitettu merkintäkieli.
11 \item[Lualatex] Latex-muotoisten lähdedokumenttien kääntäjäohjelma.
12   Sisältää Lua-nimi\-sen ohjelmointikielen.
13 \end{list}

```

⇒ TEX: Tekstidokumenttien ladontaan erikoistunut ohjelmointikieli.

LATEX: Dokumenttien kirjoittajille tarkoitettu merkintäkieli.

LUALATEX: Latex-muotoisten lähdedokumenttien kääntäjäohjelma. Sisältää Lua-nimisen ohjelmointikielen.

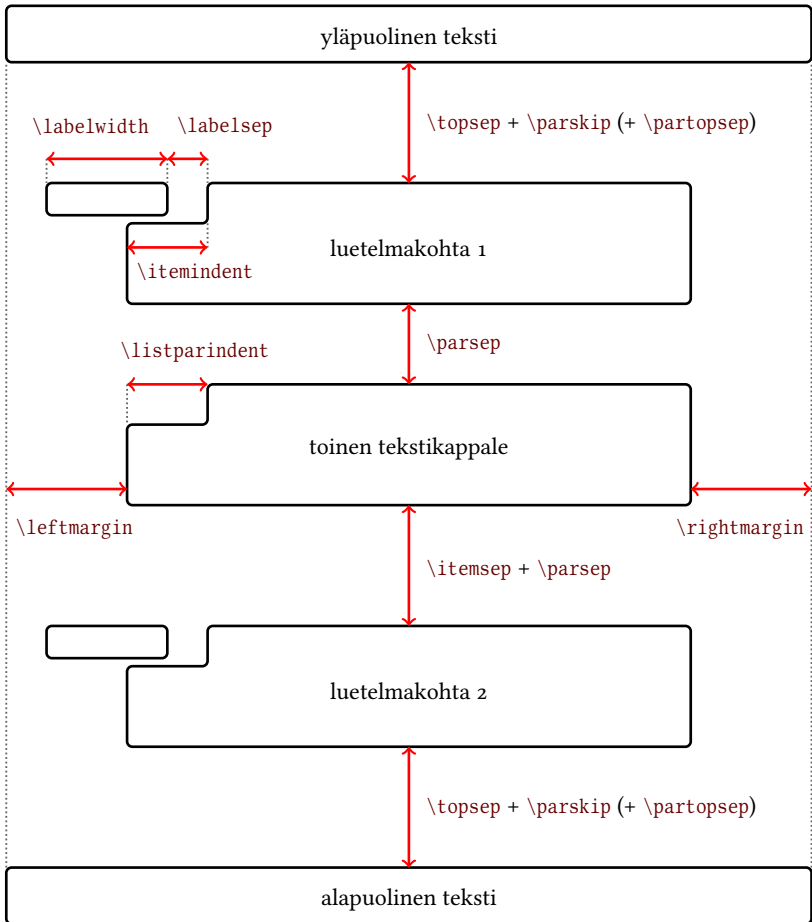
Esimerkki 4.12: Määritelmäluetelmien tekeminen `list`-ympäristön avulla. Sisäisesti komento `\makelabel` huolehtii luettelamerkkien eli tässä käsitteiden nimien latomisesta.

Kirjoittaja voi määritellä `\makelabel`-komennon uudelleen sellaiseksi kuin haluaa. Se on tarpeen esimerkiksi silloin, kun täytyy latoa kaikki luettelamerkit eri fontilla kuin muu teksti. Varsinkin määritelmäluetelmissa (luku 4.6.1), joissa luetellaan käsitteitä ja niiden määritelmiä, on usein hyödyllistä, että käsitteet erottuvat muusta tekstistä selvästi. Esimerkki 4.12 selventää tätä asiaa.

Luotelmiin liittyviä mittoja on useita, ja ne voi asettaa sopiviin arvoihin `list`-ympäristön toisen argumentin sisällä. Mittojen sijainnit on koottu kuvaan 4.2, ja seuraavassa on lisätietoa niiden merkityksestä.

`\parskip` Pystysuuntainen mitta ja väli, joka ei liity pelkästään `list`-ympäristöön vaan on yleinen tekstikappaleiden välinen etäisyysmitta (luku 4.1.2). Mitta on vaikuttaa kuitenkin myös `list`-ympäristöön, ja sen voi määritellä väliaikaisesti uudelleen ympäristön yhteydessä.

`\topsep` Pystysuuntainen väli, joka ladotaan ennen ja jälkeen `list`-ympäristön, yhdessä mittojen `\parskip` ja `\partopsep` kanssa.



Kuva 4.2: Luettelien tekemiseen tarkoitetun `list`-ympäristön mitat.

- `\partopsep` Pystysuuntainen väli, joka lisätään `list`-ympäristöä ennen ja sen jälkeen silloin, kun ympäristö aloittaa uuden tekstikappaleen eli sitä ennen on tyhjä rivi.
- `\parsep` Pystysuuntainen väli, joka tulee luotelman sisällä kaikkien luettelukohtien väliin sekä saman luettelukohdan eri tekstikappaleiden väliin.
- `\itemsep` Pystysuuntainen väli, joka tulee luotelman sisällä luettelukohtien väliin.
- `\leftmargin` Luotelman kappaleiden vasen sisennys.
- `\rightmargin` Luotelman kappaleiden oikea sisennys. Oletusarvo on nolla.
- `\labelwidth` Luettelamerkin leveys. Merkki ladotaan oletuksena tämän alueen oikeaan reunaan, mutta se riippuu `\makelabel`-komenton on määrittelystä. Jos luettelamerkki on leveämpi kuin `\labelwidth`-mitta, luettelamerkin jälkeinen teksti siirtyy oikealle siten, että merkki mahtuu kokonaan. Luettelamerkin alueen vasen reuna sijaitsee kohdassa, jonka voi laskea seuraavasta kaavasta: $\leftmargin - \labelwidth + \itemindent - \labelsep$. Mikäli haluaa, että luettelamerkin vasen reuna on sisennyksen nollakohdassa, täytyy edellä mainitut mitat asettaa siten, että mittojen yhteistulos on nolla. Esimerkissä 4.12 on tehty juuri näin.
- `\labelsep` Vaakasuuntainen väli luettelamerkin jälkeen, ennen luotelman sisältötekstiä. Oletusarvo on 0,5 em.
- `\itemindent` Ylimääräinen luettelukohtien ensimmäisen rivin sisennys. Tätä voi tarvita, kun haluaa säätää luettelamerkin vasemman reunan tiettyyn kohtaan. Katso mitta `\labelwidth` edellä. Oletusarvo on nolla.
- `\listparindent` Luettelukohtien toisen ja sitä seuraavien tekstikappaleiden ensimmäisen rivin sisennys. Oletusarvo on nolla.

Kun on lopulta saanut `list`-ympäristön asetukset kohdalleen ja luettelmat ovat sellaisia kuin pitääkin, on varmaankin hyvä aika luoda oma ympäristö, joka piilottaa asetukset yhden ympäristömäärittelyn sisään. Omaa ympäristöä on sitten helppoa käyttää useitakin kertoja omassa lähdedokumentissa.

```

1 \newenvironment{numeroimaton}[1][\textbullet]{%
2   \begin{list}{#1}{
3     \setlength{\leftmargin}{1.1em}
4     \setlength{\labelsep}{.2em}
5     \setlength{\itemsep}{.4ex plus .1ex}
6     \setlength{\parsep}{.2ex}
7   }
8 }{\end{list}}

```

Esimerkki 4.13: Oman numeroimattoman luettelman tekeminen `list`-ympäristön avulla.

```

1 \newenvironment{numeroitu}[1][0]{%
2   \begin{list}{\arabic{enumi}.}{
3     \usecounter{enumi}
4     \setcounter{enumi}{#1}
5     \renewcommand{\makelabel}[1]{\hfill\bfseries\large #1}
6     \setlength{\leftmargin}{2em}
7     \setlength{\labelsep}{.5em}
8     \setlength{\itemsep}{.4ex plus .1ex}
9     \setlength{\parsep}{.2ex}
10  }
11 }{\end{list}}

```

Esimerkki 4.14: Oman numeroidun luettelman tekeminen `list`-ympäristön avulla.

Esimerkissä 4.13 on vinkkejä oman numeroimattoman luettelman tekemiseen. Luettelma on nimeltään `numeroimaton`, ja sille voi antaa hakusulkeissa valinnaisen argumentin, jolla valitaan luettelmamerkki. Oletuksena se on luettelmaympyrä (`\textbullet`).

Esimerkki 4.14 tekee numeroidun ympäristön nimeltä `numeroitu`. Tälle ympäristölle voi antaa valinnaisen argumentin, jolla ilmaistaan numerointilaskurin alkukohta. Oletuksena se on nolla, eli luettelma-kohtien numerointi alkaa luvusta 1. Tämä ympäristö määrittelee myös `\makelabel`-komennon siten, että luettelmanumerot ladotaan lihavoituna (`\bfseries`) ja suuremmalla fontilla (`\large`). Numerot myös tasataan luettelmamerkille varatun tilan oikeaan reunaan (`\hfill`). Huomaa, että parametri `#1` kuuluu `numeroitu`-ympäristön määrittelmään ja paramet-

ri ##1 kuuluu sisempään, `\makeLabel`-komennon määritelmään. Katso lisätietoa komentojen määrittämisestä käsittelevästä luvusta 2.2.2.

4.7 Taulukot

Taulukoiden tekemiseen on perus Latexissa kaksi ympäristöä, `tabular` ja `tabular*`, joskin jälkimmäisen tilalle sopii yleensä paremmin `tabularx`-paketin ympäristö `tabularx`. Ennen kuin käsitellään taulukoiden teknisiä ohjeita on kuitenkin syytä puhua niiden sijoittelusta ja typpografiasta.

Taulukon voi sijoittaa dokumenttiin eri tavoin. Yksi mahdollisuus on sijoittaa ne normaalin tekstivirran sekaan omiksi tekstikappaleikseen. Pieni pystysuuntainen väli on silloin tarpeen ennen ja jälkeen taulukon. Jos tekstikappaleiden välissä ei ole normaalisti väliä (`\parskip`), täytyy sellainen lisätä käsin. Pystysuuntaisia välejä käsitellään luvussa 4.1.2. Taulukolle tulee oletuksena sama sisennys kuin tekstikappaleiden ensimmäisellä rivillä on eli mitta `\parindent` (luku 4.1.3). Jos sisennyksen haluaa pois, täytyy käyttää komentoa `\noindent`.

Toinen tyypillinen vaihtoehto on sijoittaa taulukko vaakasuunnassa sivun keskelle, jolloin se erottuu muusta tekstistä vielä selvemmin. Tämän voi toteuttaa kirjoittamalla taulukkoympäristö `center`-ympäristön sisään (4.1.1).

Kolmas vaihtoehto on tehdä taulukosta leijuva eli antaa Latexin sijoittaa se sopivaan paikkaan. Samalla taulukolle annetaan kuvateksti ja yksilöllinen tunniste, niin että siihen voi viitata tekstistä. Leijuvia osia ja ristiviitteitä käsitellään luvuissa 4.9 ja 4.10.

Taulukkojen ulkoasua ja helppolukuisuutta käsittelevät ohjeet yleensä neuvovat, että pystyviivoja ei pitäisi juuri käyttää. Ne häiritsevät vasemmalta oikealle lukemista ja vaikeuttavat katseen tarkentamista solujen sisältöön. Tavallinen sarakkeiden välinen tyhjä tila on yleensä riittävä erottamaan solut toisistaan.

Se on hyvä yleisohje, mutta sitä tuskin kannattaa pitää ihan ehdottomana sääntönä. Joskus on tarpeen jakaa sarakkeet mielekkäisiin ryhmiin ja nimenomaan ohjata katsetta pystysuunnassa tiettyjä sarakkeita pitkin. Katseen ei välttämättä haluta lipsuvan tietyn kohdan yli, ja

Tyyppi	Merkitys
l	vasemmalle tasattu sarake (left)
c	keskitetty sarake (center)
r	oikealle tasattu sarake (right)
p{m}	sarake, jonka leveys on mitta m
	sarakkeiden välissä pystyviiva
@{...}	sarakkeiden välissä olevaa tekstiä tai komentoja
*{n}{s}	toistetaan n kertaa sarakkeet s

Taulukko 4.6: Taulukoiden saraketyyppien ja sarakkeiden välien määrittely.

pystyviiva sopii hyvin siihen tarkoitukseen. Tämän luvun esimerkeissä käytetään paljon vaaka- ja pystyviivoja, jotta lukija hahmottaa helposti, miten esimerkkikoodin solut ovat yhteydessä ladottuun taulukkoon. Käytännössä viivoja ei kannata käyttää niin paljon, vaan pari vaakaviivaa yleensä riittää.

4.7.1 Perustoiminnot

Taulukoiden perusympäristö on `tabular`, jolle täytyy antaa ainakin yksi argumentti. Ympäristön rakenne on seuraavanlainen:

```
\begin{tabular}[sijainti]{sarakkeet}
  solu 1 & solu 2 \\
  solu 3 & solu 4 \\
\end{tabular}
```

Ympäristön valinnainen argumentti `sijainti` määrittelee taulukkoympäristön sijainnin pystysuunnassa, jos se on samassa kappaleessa muun tekstin kanssa. Oletusarvo on `c` (center), joka tarkoittaa, että pystysuunnassa taulukon peruslinja on sen keskellä. Muut vaihtoehdot ovat `t` (top), eli taulukon peruslinja on yläreunassa, ja `b` (bottom), eli taulukon peruslinja on sen alareunassa.

Pakollinen argumentti `sarakkeet` määrittelee taulukon sarakkeiden määrän ja tyypin. Ne ilmaistaan tietyillä kirjaimilla tai muilla merkeillä, jotka on koottu taulukkoon 4.6. Taulukon neljä ensimmäistä tyyppiä `lcrp` määrittelevät, millaisesta sarakkeesta on kyse. Syntyvässä taulukossa tulee olemaan niin monta saraketta kuin näitä on ympäristön `sarakkeet`-argumentissa. Kolme ensin mainittua luovat sarakkeen, jon-


```

1 \begin{tabular}{|l@{\hspace{4em}}r|c|}
2 \hline
3 Vasen & Oikea & Keskitetty \\ \ [.5ex]
4 \hline
5 16 & 32 & 11732 \\
6 \cline{1-2}
7 71 & 87235 & 2 \\
8 \cline{3-3}
9 1235 & 238 & 982 \\
10 \hline
11 \end{tabular}

```

⇒

Vasen	Oikea	Keskitetty
16	32	11732
71	87235	2
1235	238	982

Esimerkki 4.15: Taulukoiden erilaisia saraketyyppejä ja viivoja.

ka leveys määräytyy sarakkeen leveimmän solun perusteella. Neljäs eli saraketyyppi p tarvitsee aaltosulkeissa argumentiksi mitan (luku 2.4), ja sillä ilmaistaan sarakkeen kiinteä leveys. Tässä saraketyypissä teksti tasataan solun molemmista reunoista eli sanavälit voivat venyä.

Merkit | ja @ voi sijoittaa edellä mainittujen saraketyyppien väliin tai ennen ensimmäistä saraketta tai viimeisen sarakkeen jälkeen. Ensin mainittu tekee sarakkeiden väliin pystyviivan. Viiva yltää taulukon kaikille riveille, mutta komennolla `\multicolumn` voi tehdä rivikohtaisia poikkeuksia. Tätä komentoa käsitellään myöhemmin. Merkki @ tarvitsee argumentin aaltosulkeissa. Se voi olla mitä hyvänsä tekstiä tai komentoja, jotka halutaan suorittaa joka rivillä sarakkeiden välissä.

Sarakemerkki * on vain keino toistaa tiettyjä sarakemäärityksiä useamman kerran. Se tarvitsee aaltosulkeissa kaksi argumenttia: ensimmäinen on luku, ja toinen sisältää mitä hyvänsä edellä mainittuja saraketyyppejä. Nämä sarakkeet toistuvat luvun ilmaiseman määrän. Esimerkiksi `*{3}{r1|}` tarkoittaa samaa kuin `r1|r1|r1|`.

Esimerkki 4.15 havainnollistaa taulukon ominaisuuksia ja eri saraketyyppejä. Sitä tuskin kannattaa pitää typografisena esimerkkinä, mut-

ta tarkoituksena on osoittaa eri saraketyyppien toiminta käytännössä. Sarakkeiden välissä on yleensä pystyviiva, mutta yhdessä välissä on @{...}-merkeillä määritelty komento `\hspace`, jolla tehdään poikkeuksellisen leveä (4 em) tyhjä tila sarakkeiden väliin.

Rivillä olevat peräkkäiset solut erotetaan toisistaan &-merkillä, ja taulukon rivi täytyy päättyä aina `\\`-komentoon. Tälle rivinvaihtokomennolle voi antaa hakasulkeissa yhden argumentin. Se on mitta ja tarkoittaa, kuinka paljon ylimääräistä pystysuuntaista tyhjää tilaa halutaan rivin jälkeen, esimerkiksi `\\[.5ex]`.

Vaakasuuntainen, koko taulukon levyinen viiva tehdään komennolla `\hline` ja määräämittäisiä viivoja komennolla `\cline`. Tälle komennolle täytyy antaa yksi argumentti, joka sisältää kaksi yhdysmerkillä erotettua lukua. Luvuilla ilmaistaan, mistä sarakkeesta mihin sarakkeeseen viiva ylittää.

4.7.2 Asetuksia

Taulukon rivit saatetaan latoa oletuksena vähän liian lähelle toisiaan – varsinkin jos käytetään vaakaviivoja. Sen korjaamiseksi on taulukoille olemassa oma rivivälikerroin, joka on komennossa `\arraystretch`. Kerroin on desimaaliluku, jolla normaali rivikorkeus kerrotaan taulukoiden sisällä. Sopiva kerroin lienee yleensä 1–1,3. Oletusarvo on 1. Kerroinkomennon voi määritellä milloin hyvänsä uudelleen, ja se tehdään seuraavan esimerkin mukaisesti:

```
\renewcommand{\arraystretch}{1.3}
```

Sarakkeiden määrittelyssä voi kirjoittaa useita pystyviivoja eli `|`-merkkejä peräkkäin. Silloin pystyviivoja myös ladotaan taulukkoon useita vierekkäin. Ladottujen viivojen väliseen etäisyyteen voi vaikuttaa mitan `\doublerulesep` avulla. Sen oletusarvo on 2 pt. Kaikkien viivoja paksuus puolestaan on mitassa `\arrayrulewidth`, jonka oletusarvo on 0,4 pt.

Jos sarakkeiden välissä ei ole @-merkillä tehtyjä poikkeuksia, ladotaan joka solun vasempaan ja oikeaan reunaan väli, joka on määritelty mitassa `\tabcolsep`. Tämän mitan suuruinen väli siis ladotaan kerran taulukon vasempaan ja oikeaan reunaan ja kaksi kertaa jokaisen sarakkeen väliin. Oletusarvo on 6 pt. Seuraavassa on koonti edellä mainittujen mittojen asettamisesta ja niiden oletusarvot.

```

\setlength{\doublerulesep}{2pt}
\setlength{\arrayrulewidth}{.4pt}
\setlength{\tabcolsep}{6pt}

```

Sarakemääritelmässä voi @-merkillä säätää sarakkeiden välit haluamansa laiseksi. Merkeillä @{} sarakeväli poistetaan kokonaan. Jos @{...}-määrittelyyn halutaan mukaan normaali sarakkeiden pystyviiva, täytyy käyttää komentoa `\vline`, esimerkiksi seuraavasti:

```

\begin{tabular}{|l@{\hspace{1em}}\vline\hspace{1em}}l|}
  ensimmäinen & toinen \\
\end{tabular}

```

4.7.3 Poikkeuksellisia sarakkeita ja rivejä

Taulukkoympäristön argumenteissa määritellään taulukon sarakkeiden määrä ja tyypit, mutta niihin on mahdollista tehdä yksittäisiä poikkeuksia taulukon sisällä. Sarakekohtaiset poikkeukset tehdään komennolla `\multicolumn` ja rivikohtaiset poikkeukset komennolla `\multirow`, joka kuuluu pakettiin `multirow`¹⁹.

Esimerkissä 4.16 on taulukoitu lukuja, jotka halutaan latoa samaan linjaan desimaalipilkun kohdalta. Tasaamisessa hyödynnetään kahden sarakkeen rajakohtaa: luvun kokonaislukuosa ja desimaaliosa tasataan tätä rajakohtaa vasten – toinen oikealle (r), toinen vasemmalle (l). Sarakeväli on poistettu tästä kohdasta kokonaan @{}-merkeillä.

Esimerkkitaulukon otsikko ”Tulokset” halutaan kuitenkin sijoittaa desimaalilukujen koko leveydelle eli kahden sarakkeen päälle. Lisäksi otsikkoriviltä halutaan poistaa pystyviivat, jotka muutoin ladotaan taulukon reunoille. Näihin asioihin tarvitaan komentoa `\multicolumn`, joka on tarkoitettu poikkeuksellisten sarakemäärittelysten tekemiseen. Komentoa käytetään seuraavasti:

```

\multicolumn{n}{sarakeet}{teksti}

```

Komennon `\multicolumn` voi sijoittaa mihin hyvänsä taulukon soluun. Komennon ensimmäinen argumentti n on luku ja ilmaisee, kuinka monen sarakkeen alueeseen vaikutetaan. Toinen argumentti sarakeet sisältää uuden sarakemäärittelyn tälle alueelle, ja se tehdään taulukon

¹⁹ <https://www.ctan.org/pkg/multirow>

```

1 \begin{tabular}{|r@{}l|}
2 \multicolumn{2}{l}{Tulokset} \\
3 \hline
4 425 & ,34 \\
5 4 & ,021 \\
6 32 & \\
7 0 & ,75 \\
8 \hline
9 \end{tabular}

```

⇒ Tulokset

425,34
4,021
32
0,75

Esimerkki 4.16: Desimaalilukujen tasaaminen pilkun kohdalta, ja poikkeuksellisten sarakkeiden tekeminen `\multicolumn`-komennolla.

4.6 merkkien avulla. Kolmas argumentti on teksti, joka ladotaan tähän uudella tavalla määritellyyn alueeseen.

Edellä on käsitelty vain soluja, joissa on yksirivinen teksti, mutta tiettyissä tilanteissa yhteen soluun voidaan latioa useampia rivejä. Sarake-tyyppi p (taulukko 4.6) on leveydeltään kiinteä, ja jos solun sisältö ei mahdu kyseiseen tilaan, se jaetaan useammalle riville ja sanoja mahdollisesti katkaistaan tavurajojen kohdalta. Myös `\newline`-komennolla voi vaihtaa riviä yksittäisen solun sisällä. Sekin toimii vain kiinteäleveyksisillä p-tyyppin sarakkeilla.

Seuraavassa on esimerkki, kuinka yhden solun sisältö voi jakautua kahdelle riville. Teknisessä mielessä taulukkorivien määrä ei kuitenkaan kasva, vaan muihin saman taulukkorivin soluihin tulee tyhjää tilaa.

```

\begin{tabular}{|l|p{6.5em}|l|}
\hline solu & Tämä ei mahdu yhdelle riville. & solu \\
\hline
\end{tabular}

```

⇒

solu	Tämä ei mahdu yhdelle riville.	solu
------	--------------------------------	------

Joskus puolestaan halutaan, että yksi solu levittäytyy useamman rivin alueelle verrattuna taulukon muihin riveihin. Tämän toteutukseen tarvitaan `multirow`-pakettia ja sen `\multirow`-komentoa. Komennon pakolliset argumentit ovat seuraavat:

```
\multirow{n}{leveys}{teksti}
```

Ensimmäinen argumentti `n` on luku, joka ilmaisee, kuinka monen rivin alueelle solu levitetään. Toinen argumentti `leveys` on tekstin leveyttä ilmaiseva mitta. Argumentiksi voi kirjoittaa pelkän tähden (`*`), jolloin käytetään tekstin luonnollista leveyttä. Kolmas argumentti on teksti, joka ladotaan tähän pystysuunnassa laajennettuun soluun.

Komennolla on myös valinnaisia argumentteja, jotka ilmaistaan hakusulkeiden avulla. Tässä niistä tärkeimmät:

```
\multirow[sijainti]{n}{leveys}[pysty]{teksti}
```

Edellisen esimerkin valinnainen argumentti `sijainti` on kirjain, jolla ilmaistaan solun sisällön pystysuuntainen asemointi ylös, keskelle tai alas: `t` (top), `c` (center, oletus) tai `b` (bottom). Valinnainen argumentti `pysty` on mitta, jolla hienosäädetään solun sisällön sijaintia pystysuunnassa. Positiivinen mitta siirtää sisältöä ylöspäin, negatiivinen alaspäin.

Komennon muista mahdollisuuksista kannattaa lukea `multirow`-paketin ohjekirjasta. Esimerkki 4.17 havainnollistaa komennon peruskäyttöä. `\multirow`-komento kirjoitetaan siihen soluun, josta monirivinen solu alkaa. Sen alapuolelta täytyy jättää riittävä määrä soluja tyhjäksi.

4.7.4 Kiinteälevyiset taulukot

Latexin toinen taulukkoympäristö `tabular*` toimii muuten samalla tavalla kuin `tabular`, mutta se tarvitsee yhden argumentin enemmän, taulukon leveysmitan:

```
\begin{tabular*}{leveys}[sijainti]{sarakkeet}
...
\end{tabular*}
```

Valitettavasti taulukon leveysmittaa voi hyödyntää vain rajallisesti. Se vaikuttaa taulukon leveyteen silloin, kun sarakkeiden välit on määritellyt venyviksi käyttämällä seuraavanlaista sarakkevälin määrittystä:

```

1 \begin{tabular}{|1|1|1|}
2 \hline
3 \multirow{2}{*}{kaksirivinen solu} & tavallisia & soluja \\
4 \cline{2-3}
5 & kahdella & rivillä \\
6 \hline
7 \end{tabular}

```

⇒

kaksirivinen solu	tavallisia	soluja
	kahdella	rivillä

Esimerkki 4.17: `multirow`-paketin `\multirow`-komennolla voi levittää yhden solun useamman rivin alueelle.

`@{\extracolsep{\fill}}`

Edellisestä on tärkeää huomata, että sarakkeiden leveydet eivät veny vaan ainoastaan niiden välissä oleva tyhjä tila. Usein olisi kuitenkin hyödyllisempää, että joidenkin sarakkeiden leveys olisi venyvä ja ne mukautuisivat koko taulukon leveyteen. Tämä ominaisuus saadaan `tabularx`²⁰-paketin avulla. Se määrittelee uuden taulukkoympäristön `tabularx`, joka on käytännössä parempi versio `tabular*`-ympäristöstä. Sen käyttö on samanlaista:

```

\begin{tabularx}{leveys}[sijainti]{sarakkeet}
...
\end{tabularx}

```

Ympäristö `tabularx` sisältää yhden uuden saraketyypin: `X` (vrt. taulukko 4.6). Tämä saraketyyppi toimii kuten tyyppi `p`, mutta sille ei anneta argumentiksi mittaa, vaan sen leveys on äärettömästi venyvä. Tämän tyyppiset sarakkeet täyttävät siis kaiken vapaana olevan tilan, ja jos niitä on useita, ne ovat oletuksena keskenään yhtä leveitä. Tämä ilmenee seuraavasta esimerkistä:

```

\begin{tabularx}{\linewidth}{|1|X|X|}
\hline luonnollinen & venyvä & venyvä \\
\end{tabularx}

```

²⁰ <https://www.ctan.org/pkg/tabularx>

⇒	luonnollinen	venyvä	venyvä
---	--------------	--------	--------

Saraketyypissä X teksti tasataan oletuksena solujen molemmista reunoista, mutta siihen on mahdollista vaikuttaa sijoittamalla sarakemäärittelyssä $>$ -merkki ja sen aaltosulkeissa olevat argumentit ennen X -merkkiä. Argumentiksi kirjoitetaan sopivia asetuskomentoja. Seuraava sarakemäärittely tekee vasemmalle tasatun X -sarakkeen:

```
>\raggedright\arraybackslashX
```

Sarakkeen solujen tasaaminen vasemmalle tulee komennon `\raggedright` vaikutuksesta (luku 4.1.1). Sen perään tarvitaan myös komento `\arraybackslash`, koska `\raggedright` määrittelee `\\`-rivinvaihtokomennon uudelleen tavalla, joka on ristiriidassa taulukon rivien lopussa tarvittavan vastaavan komennon kanssa. `\arraybackslash` palauttaa määritelmän sellaiseksi, että se toimii taulukossa.

`tabularx`-paketin myötä voi määritellä omiakin saraketyyppejä. Hyödyllisiä tyyppiejä voisivat olla esimerkiksi LCR, jotka toimisivat lähes samoin kuin perus `Latexin` saraketyypit `lcr` (taulukko 4.6) mutta olisivat kuitenkin leveydeltään venyviä. Omia saraketyyppejä tehdään komennolla `\newcolumntype`, ja edellä mainitut tyytit voisi määritellä seuraavasti:

```
\newcolumntype{L}{>\raggedright\arraybackslashX}
\newcolumntype{C}{>\centering\arraybackslashX}
\newcolumntype{R}{>\raggedleft\arraybackslashX}
```

Mitä hyvänsä saraketyyppejä voi tehdä vastaavalla tavalla. Esimerkiksi taulukon ensimmäiseen, otsikoita ilmaisevaan saraketyyppiin 0 voitaisiin laittaa fonttikomento, jotta koko sarake ladotaan automaattisesti eri kirjainleikkauksella kuin muu taulukko.

```
\newcolumntype{0}{>\bfseries}l % lihavoitu otsikkosarake
```

Ympäristön `tabularx` käyttöön liittyy tiettyjä rajoituksia: sitä ei voi käyttää normaalisti omien ympäristöjen määrittelyssä (luku 2.3). Esimerkiksi seuraava ympäristön määrittely ei käytännössä toimi:

```
\newenvironment{omataulukko}[1]
{\begin{tabularx}{\linewidth}{#1}}
{\end{tabularx}}
```

Edellä mainitun puutteen saa korjattua käyttämällä oman ympäristön määrittelyssä `tabularx`-ympäristön aloitukseen komentoa `\tabularx` ja lopetukseen komentoa `\endtabularx`. Ei siis käytetä normaaleja aloitus- ja lopetuskomentoja `\begin` ja `\end`.

```
\newenvironment{omataulukko}[1]
{\tabularx{\linewidth}{#1}}
{\endtabularx}
```

4.7.5 Muita paketteja

`tabularx`-paketin kanssa samankaltainen on paketti `tabulary`²¹, jossa on joitakin erilaisia ominaisuuksia. Se muun muassa sisältää valmiiksi uudet saraketyypit LCRJ.

Edellä esitellyt taulukkoympäristöt ladotaan aina kokonaisuena eli vain yhden sivun alueelle. Useammalle sivulle jakautuvia taulukkoita voi tehdä `longtable`²²-paketin taulukkoympäristön `longtable` avulla. Sitä voi käyttää kuten `tabular`-ympäristöä, mutta lisäksi se sisältää komentoja, joiden avulla saa joka sivulla toistumaan automaattisesti tietyt otsikkorivit tai yhteenvetorivit. Toinen mahdollisuus yli sivurajan yltyvien taulukon kaltaisten rakenteiden tekemiseen ovat sarkaimet, joita käsitellään luvussa 4.8.

Hyödyllinen taulukoiden apupaketti on `booktabs`²³, jonka avulla saa erilaisia ja tyylikkäämpiä vaakaviivoja. Paketin komennot `\toprule`, `\midrule`, `\bottomrule` ja `\cmidrule` piirtävät rivien väliin erivahvuisia viivoja ja huolehtivat myös niiden ylä- ja alapuolisesta välistä paremmin kuin Latexin omat viivakomennot.

Taulukon rivien taustan voi värittää `xcolor`²⁴-paketin (luku 4.2.7) komentojen avulla. Varsinkin leveiden taulukoiden lukemista voi helpottaa, jos rivien eri taustavärit vuorottelevat. Tarkemmat tiedot selviävät paketin ohjekirjasta, mutta seuraavan esimerkin avulla ominaisuutta pääsee kokeilemaan:

```
\usepackage[table]{xcolor} % lähdedokumentin esittelyosaan
\rowcolors{1}{blue!10}{red!10} % ennen taulukkoympäristöjä
```

²¹ <https://www.ctan.org/pkg/tabulary>

²² <https://www.ctan.org/pkg/longtable>

²³ <https://www.ctan.org/pkg/booktabs>

²⁴ <https://www.ctan.org/pkg/xcolor>

4.8 Sarkaimet

Sarkaimet ovat ennalta määritettyjä vaakasuuntaisia sisennys- tai tasmauskohtia. Niitä voi hyödyntää tekstin tasaamisessa tiettyihin kohtiin rivillä. Suomenkielisissä asiakirjoissa sarkainkohdat ovat 23 millimetrin välein, mutta jos ei ole kirjoittamassa virallista asiakirjaa, voi kirjoittaja määrittää sarkainkohdat niin kuin haluaa.

Ajatus sarkaimista on vanha, sillä mekaanisissa kirjoituskoneissakin oli sarkainnäppäin (`tab`, `tabular`), joka siirsi telaa eteenpäin seuraavaan sarkainkohtaan. Se helpotti tekstin tasaamista tiettyyn kohtaan ja sen myötä taulukoiden kirjoittamista.

Latexissa on taulukoita varten omatkin ympäristönsä (luku 4.7), mutta jos ei tarvitse varsinaista taulukkoa vaan ainoastaan yksittäisiä tasmauskohtia, voi olla kätevämpää käyttää `tabbing`-sarkainympäristöä tai rakentaa tasaukset näkymättömien laatikoiden avulla. Näitä käsitellään seuraavissa alaluvuissa.

4.8.1 Sarkainympäristö

Sarkaimet voi toteuttaa niitä varten tehdyn ympäristön eli `tabbing`-ympäristön avulla. Esimerkki 4.18 havainnollistaa ympäristön peruskäyttöä. Rivillä 2 asetetaan kaikki sarkainkohdat eli tehdään vaakasuuntaisia välejä `\hspace`-komennolla ja merkitään sarkainkohdat `\=`-komennolla. Sen rivin lopussa on komento `\kill`, joka hylkää rivin eli jättää sen latomatta. Seuraavien rivien lopussa on `\-`komento – paitsi viimeisellä rivillä ennen ympäristön lopetuskomentoa. Rivin sisällä seuraavaan sarkainkohtaan hypätään `\>`-komennolla.

`tabbing`-ympäristön sisällä on käytettävissä muutama muukin erityiskomento, joilla vaikutetaan sarkainten käsittelyyn. Seuraavassa kerrotaan ympäristön kaikkien erityiskomentojen merkitys, ja esimerkiksi 4.19 havainnollistetaan joidenkin kommentojen käyttöä.

`\-` Rivinvaihto, joka täytyy kirjoittaa jokaisen rivin loppuun, lukuun ottamatta viimeistä riviä ennen ympäristön lopetuskomentoa `\end`.

`\kill` Rivinvaihto, joka hylkää kyseisen rivin. Tämä on käyttökelpoinen sarkainkohtien määrittelyrivillä, jota itsessään ei haluta latoa näkyviin.


```

1 \begin{table}
2   \begin{center}
3     \renewcommand{\arraystretch}{1.3}
4     \begin{tabular}{ll}
5       Upeita & Lukuja \\
6       \hline
7       324 & 33 \\
8       2 & 49 \\
9       \hline
10    \end{tabular}
11  \end{center}
12  \caption{Upea kuvateksti.}
13 \end{table}

```

Esimerkki 4.21: Leijuvan taulukon toteuttaminen `table`-ympäristön avulla. Varsinainen taulukko syntyy ympäristön `tabular` avulla.

merkki”, jota käytetään leijuvien Latex-koodiesimerkkien toteuttamiseen.

Koska Latex huolehtii leijuvien osien sijoittamisesta, niiden lopullinen järjestyskin voi hieman poiketa siitä, miten ne ovat lähdetiedostossa. Tosin samantyyppiset leijuvat osat, esimerkiksi kuvat, ladotaan aina siinä järjestyksessä kuin ne ovat lähdetiedostossa. Sen sijaan erityyppiset leijuvat osat saatetaan latoa eri järjestyksessä: esimerkiksi jokin taulukko saatetaan latoa ennen tiettyä kuvaa, vaikka ne olisivat lähdedokumentissa päinvastaisessa järjestyksessä. Latex pyrkii latomaan hyvännäköisiä sivuja, ja leijuvien osien sijoittelulla se voi vaikuttaa asiaan.

4.9.1 Leijuvat taulukot ja kuvat

Käytännössä leijuvat osat ovat ympäristöjä, joiden sijoittelusta vastaa Latex eikä kirjoittaja. Tosin kirjoittajakin voi hieman vaikuttaa asiaan, ja tätä käsitellään luvussa 4.9.4. Ympäristö `table` on tarkoitettu taulukoille, ja ympäristö `figure` on kuville. Esimerkistä 4.21 ilmenee leijuvan taulukon toteuttamisen perusasiat. Muuntotyypiset leijuvat osat tehdään samalla tavalla, ympäristö on vain eri.

Latex ei sinänsä ota kantaa siihen, mitä leijuva ympäristö sisältää, eli `table`-ympäristön sisällä ei tarvitse olla taulukkoympäristöä (luku

4.7), eikä `figure`-ympäristön sisällä ole pakko olla kuvaa (luku 4.15). Valitut ympäristöt vaikuttavat kuitenkin kuvatekstiin: `table`-ympäristön kuvatekstiin tulee suomen kieliasetuksilla sana ”Taulukko”, ja `figure`-ympäristössä se on ”Kuva”. Nimet tulevat komennoista `\tablename` ja `\figurename`, jotka kirjoittaja voi määrittää uudelleen. Se kannattaa tehdä dokumentin esittelyosassa seuraavalla tavalla:

```
\addto{\captionsfinnish}{
  \renewcommand{\tablename}{Hökötys}
  \renewcommand{\figurename}{Himmeli}
}
```

Edellä käytettiin `\addto`-komentoa, joka kuuluu kielipaketteihin `polyglossia` ja `babel`. Sillä lisätään omia komentoja tietyn kielen asetuksiin, tässä suomen kieliasetuksiin (`\captionsfinnish`). Kieliasetuksia käsitellään tarkemmin luvussa 3.5.

Kuvatekstit tehdään komennolla `\caption`, joka sijoitetaan leijuivan ympäristön sisälle, kuten esimerkissä 4.21 on tehty. Komennolle annetaan yksi argumentti eli haluttu kuvateksti. Siihen ei pidä kirjoittaa leijuivan osan tyyppiä eikä numeroa (esim. ”Taulukko 3”), sillä nämä Latex tekee automaattisesti. `\caption`-komennolle voi antaa myös valinnaisen argumentin, jolla ilmaistaan kuvatekstistä lyhempi versio:

```
\caption[Lyhyt kuvateksti.]{Varsinainen pitkä kuvateksti.}
```

Varsinainen (pitkä) kuvateksti ladotaan aina leijuivan osan yhteyteen ja lyhempää versiota käytetään mahdollisesti luettelossa (luku 4.9.3) ja ristiviitteissä. Kuvatekstin eli `\caption`-komennon jälkeen leijuvaan ympäristöön lisätään usein `\label`-komennolla yksilöllinen tunniste ristiviitasta varten. Komentoa käsitellään ristiviitteiden yhteydessä luvussa 4.10.

Leijuvista ympäristöistä on olemassa myös tähdelliset versiot, `table*` ja `figure*`. Ympäristöjen eri versioilla on merkitystä vain kaksipalstaisessa tilassa (luku 4.12), jossa normaaliversio (esim. `table`) ladotaan yhden palstan sisään ja tähdellinen versio (esim. `table*`) sijoitetaan palstojen ulkopuolelle, koko sivun tilaan.

Valitsin	Merkitys
name	kuvateksteissä näkyvä tyyppi
placement	sijoitteluasetukset
fileext	väliaikaistiedoston päätte
within	ylemmäntasoinen laskuri

Taulukko 4.7: `\DeclareNewFloatType`-komennon valitsimia, joilla vaikutetaan leijuvan osan ominaisuuksiin.

```

1 \DeclareNewFloatType{esimerkki}{
2   name=Esimerkki,
3   placement=tbp,
4   fileext=loesim,
5   within=chapter
6 }
```

Esimerkki 4.22: `\DeclareNewFloatType`-komennon käyttö.

4.9.2 Muut leijuvat osat

Omia, itse nimettyjä leijuvia osia tehdään `floatrow`²⁵-paketin avustuksella. Se sisältää komennon `\DeclareNewFloatType`, jonka argumentit ovat seuraavanlaiset:

```
\DeclareNewFloatType{tyyppi}{valitsimet}
```

Komennon ensimmäinen argumentti tyyppi on leijuvan osan tyyppi ja käytännössä myös ympäristön nimi. Jos tyyppiksi antaa sanan `esimerkki`, syntyy leijuvat ympäristöt `esimerkki` ja `esimerkki*`. Argumentin `valitsimet` avulla voi vaikuttaa tarkemmin leijuvan osan ominaisuuksiin. Mahdolliset valitsimet on koottu taulukkoon 4.7. Eri valitsimet erotetaan toisistaan pilkulla, eli komentoa käytetään esimerkin 4.22 tavoin.

Valitsimen `name` arvo näkyy kuvateksteissä leijuvan osan tyyppinä. Valitsimella `placement` tehdään sijoitteluasetuksia, joita käsitellään luvussa 4.9.4. Käytännössä ne ovat tiettyjä kirjaimia, joilla vaikutetaan leijuvan osan sijoitteluun. Oletusarvo on `tbp`.

Valitsin `fileext` määrittelee tiedoston päätteen. Nimittäin leijuvien osien luetteloita tehtäessä Latex käyttää väliaikaistiedostoa, ja sen päät-

²⁵ <https://www.ctan.org/pkg/floatrow>

teen voi valita tällä valitsimella. Luetteloissa on tapana käyttää tiedostopäätettä, joka alkaa kirjaimilla `lo` (list of). Latexin omissa leijuissa osissa päätteet ovat `lot` (list of tables) ja `lof` (list of figures). Oletuksena omille leijuville osille tulee päätteeksi `lo`, jonka perässä on osan tyyppi.

Valitsimen `within` avulla vaikutetaan leijuvien osien numeroinnin riippuvuuteen muista laskureista. Oletuksena leijuvan osan laskuri ei ole riippuvainen muista laskureista eli osat numeroidaan dokumentin alusta saakka yhdellä luvulla. Arvoksi voi antaa myös Latexin laskurin nimen: mielekkäitä arvoja ovat esimerkiksi `chapter` ja `section`. Vaikutus on se, että leijuvan osan laskuri on riippuvainen näistä laskureista ja nollautuu aina, kun toisen laskurin arvo kasvaa. Lisäksi kuvateksteissä näkyy kaksiosainen numerointi (4.1, 4.2 jne.), jossa ensimmäinen luku on ylemmätasoisien laskurin arvo – esimerkiksi kirjan pääluvun numero – ja toinen on leijuvan osan laskurin arvo. Tällaisia hierarkisia laskureita käsitellään tarkemmin luvussa 2.5.1. Leijuvan osan numeroinnin ulkoasun muokkaamista käsitellään luvussa 4.9.5.

4.9.3 Leijuvien osien luettelot

Dokumentin leijuvista osista voi latoa automaattisesti luetteloita: komennolla `\listoftables` syntyy taulukkoluetelo ja komennolla `\listoffigures` kuvaluettelo. Omien leijuvien osien (luku 4.9.2) luettelo tehdään `floatrow`-paketin komennolla `\listof`, jolle annetaan argumentiksi leijuvan osan tyyppi sekä luettelon otsikko.

```
\listoftables           % taulukkoluetelo (table)
\listoffigures         % kuvaluettelo (figure)
\listof{esimerkki}{Esimerkit} % luettelo esimerkki-tyyppisistä osista
```

Leijuvien osien luettelot muistuttavat sisällysluetteloita (luku 4.5), eli niille tulee automaattisesti otsikko kuten ”Taulukot” tai ”Kuvat” ja automaattisesti sisältö eli kaikki `\caption`-komennolla ilmaistut (lyhyet) kuvatekstit sekä tietysti sivunumerot.

Leijuvien osien luetteloiden ulkoasua voi muokata samalla tavalla kuin sisällysluetteloidenkin: käytetään pakettia `titletoc`²⁶ ja määritellään sen `\titlecontents`-komennolla luettelokohtien ulkoasu. Esimerkistä 4.23

²⁶ <https://www.ctan.org/pkg/titletoc>

1	<code>\titlecontents{table}</code>	% tyyppi: table
2	<code>[8mm]</code>	% vasen sisennys (mitta)
3	<code>{\addvspace{3bp}\rmfamily\small}</code>	% yläpuolinen koodi
4	<code>{\contentslabel{8mm}}</code>	% numeroitu kohta
5	<code>{}</code>	% numeroimaton kohta
6	<code>{~\small\titlerule*[3mm]{.}\contentspage}</code>	% pisteviiva ja sivunumero
7	<code>[]</code>	% alapuolinen koodi

Esimerkki 4.23: Leijuvien osien luettelon ulkoasua muokataan `\titlecontents`-komennolla, joka on peräisin `titletoc`-paketista.

selvinnee perusajatus, ja lisätietoa on sisällysluetteloita käsittelevässä luvussa 4.5 sekä tietysti paketin ohjekirjassa.

Latexin omien leijuvien osien luetteloiden otsikot tulevat kieliasetuksista ja komennoista `\listtablename` ja `\listfigurename`, jotka kirjoittaja voi määritellä uudestaan. Järkevää on tehdä se lähdedokumentin esittelyosassa ja käyttää `\addto`-komentoa, jolla asetukset lisätään tietynt kielen asetuksiin.

```
\addto{\captionsfinnish}{
  \renewcommand{\listtablename}{Taulukkoluettelo}
  \renewcommand{\listfigurename}{Kuvaluettelo}
}
```

Luetteloihin saa lisättyä omia rivejä samoilla keinoilla kuin sisällysluetteloihinkin eli komennolla `\addcontentsline`. Komennon ensimmäinen argumentti on luettelon tyyppiä vastaavan tiedoston päätte. Tyyppi `lot` (list of tables) on taulukoille ja tyyppi `lof` (list of figures) on kuville. Omilla leijuilla osilla (luku 4.9.2) tiedoston päätte on jotakin muuta ja valitaan osan määrittelyn yhteydessä.

```
\addcontentsline{lot}{table}{Ylimääräinen taulukko}
\addcontentsline{lof}{figure}{Ylimääräinen kuva}
```

Komennon toinen argumentti tarkoittaa luettelomerkinnän tasoa (vrt. otsikkotasot). Leijuvien osien luetteloissa on vain yksi taso, ja siihen kirjoitetaan kyseisen leijuivan osan tyyppi eli ympäristön nimi, esimerkiksi `table` tai `figure`. Komennon kolmas argumentti on luetteloon lisättävä merkintä.

Valitsin	Merkitys
t	sivun yläosaan (top)
b	sivun alaosaan (bottom)
p	leijuvien osien sivulle (page)
h	mieluiten tähän kohtaan (here)
H	ehdottomasti tähän kohtaan (here, <code>floatrow</code>)
!	ei huomioida tiettyjä sijoittelurajoituksia

Taulukko 4.8: Leijuvien osien sijoitteluun voi vaikuttaa ympäristön valinnaisen argumentin avulla. Valitsin H on `floatrow`-paketin ominaisuus.

4.9.4 Sijoittelu sivulle

Latex sijoittaa leijuvat osat yleensä sivun yläosaan, joskus alaosaan tai omalle sivulleen, jossa ei ole leipätekstiä lainkaan. Yksittäisen ympäristön sijoitteluun voi vaikuttaa valinnaisen argumentin avulla:

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

Valinnainen argumentti `sijainti` on tietty merkki tai merkkien yhdistelmä, jolla vaikutetaan leijuvan osan sijoitteluun. Taulukossa 4.8 ovat mahdolliset vaihtoehdot. Oletusarvo on `tbp`, eli sen mukaan leijuva osa pyritään sijoittamaan ensisijaisesti sivun yläosaan (t) mutta mahdollisesti myös alaosaan (b) tai omalle sivulleen (p) mahdollisesti muiden leijuvien osien kanssa.

Vaihtoehto `h` pyrkii latomaan leijuvan osan siihen kohtaan kuin se on lähdetiedostossa. Se on kuitenkin vain ehdotus eikä useinkaan toteudu: Latex lisää automaattisesti aina `h`-valitsimen perään `t:n` eli sivun yläosaan sijoittamisen. Latex käyttää sitä, jos se saa sillä tavoin omasta mielestään paremman lopputuloksen. Lataamalla paketin `floatrow`²⁷ voi käyttää myös myös valitsinta `H`. Se tarkoittaa ehdotonta vaatimusta, että juuri tähän kohtaan leijuva osa pitää sijoittaa. Tätä valitsinta ei voi yhdistää muiden valitsimien kanssa.

²⁷ <https://www.ctan.org/pkg/floatrow>

Valitsinta ! käytetään yhdessä edellä mainittujen valitsimien tbp kanssa. Se poistaa tietyt sijoitteluun vaikuttavat säännöt tai rajoitukset. Niitä käsitellään seuraavaksi.

Leijuvien osien automaattisessa sijoittelussa vaikuttaa kymmenen erilaista parametria, joihin kirjoittaja voi vaikuttaa: kolme laskuria, kolme mittaa ja neljä komentoa. Joillekin asetuksille on olemassa rinnakkainen versio, jota käytetään kahden palstan tilassa (luku 4.12). Ne koskevat leijuvien ympäristöjen tähdellisiä versiota, kuten `table*` ja `figure*`, eli tilannetta, jossa leijuva osa ei ole palstojen sisällä, vaan koko sivun leveys on käytettävissä.

Seuraavat kolme laskuria vaikuttavat sivulle ladottavien leijuvien osien enimmäismäärään. Ne koskevat tilannetta, jossa sivulla on myös tavallista leipätekstiä tai muuta sisältöä. Laskureille asetetaan uusi arvo komennolla `\setcounter` (luku 2.5). Näiden laskurien vaikutuksen voi poistaa kokonaan käyttämällä leijuvan ympäristön valitsinta ! (taulukko 4.8).

`topnumber` Sivun yläosaan sijoitettavien leijuvien osien enimmäismäärä. Oletusarvo on 2. Kahden palstan tilassa käytetään laskuria `dbltopnumber`.

`bottomnumber` Sivun alaosaan sijoitettavien leijuvien osien enimmäismäärä. Oletusarvo on 1. Kahden palstan tilassa leijuvia osia ei sijoiteta sivun alaosaan täysilevyisenä.

`totalnumber` Samalla sivulla olevien leijuvien osien enimmäismäärä, kun sivulla on myös tavallista leipätekstiä. Oletusarvo on 3.

Seuraavat kolme mittaa vaikuttavat leijuvien osien pystysuuntaisiin väleihin. Mitat asetetaan komennolla `\setlength` (luku 2.4).

`\floatsep` Samalla sivulla kahden peräkkäisen leijuvan osan pystysuuntainen väli. Oletusarvo on venyvä mitta 12pt plus 2pt minus 2pt. Kahden palstan tilassa käytetään mitta `\dblfloatsep`.

`\textfloatsep` Leijuvan osan ja leipätekstin välinen pystysuuntainen väli. Oletusarvo on venyvä mitta 20pt plus 2pt minus 4pt. Kahden palstan tilassa käytetään mitta `\dbltextfloatsep`.

`\intextsep` Pystysuuntainen väli leipätekstin ja leijuvan osan välissä silloin, kun leijuva osa sijoitetaan osaksi tekstivirtaa. Tämä koskee

sijoitteluasetuksia h ja H (taulukko 4.8). Oletusarvo vaihtelee dokumenttiluokan fonttikokoasetusten (luku 3.1.2) perusteella. Jos dokumenttiluokan valitsin on 10pt (oletus) tai 11pt, oletusmitta on 12pt plus 2pt minus 2pt. Dokumenttiluokan valitsimella 12pt oletus on 14pt plus 4pt minus 4pt.

Seuraavat neljä komentoa sisältävät pelkän desimaaliluvun 0–1. Luku kertoo leijuvien osien tai leipätekstin viemän suhteellisen tilan samalla sivulla. Komennot määrittävät uudelleen komennolla `\renewcommand` (luku 2.2.2). Näiden vaikutuksen voi poistaa kokonaan leijuva ympäristön valitsimella ! (taulukko 4.8).

`\topfraction` Sivun yläosaan sijoitettavien leijuvien osien viemä enimmäistila suhteessa sivun tekstialueen korkeuteen. Oletusarvo on 0.7, mikä tarkoittaa, että leijuvat osat voivat viedä korkeintaan 0,7-kertaisesti (70 %) sivulla olevan pystysuuntaisen tilan. Kahden palstan tilassa käytetään komentoa `\dbltopfraction`.

`\bottomfraction` Sivun alaosaan sijoitettavien leijuvien osien viemä enimmäistila suhteessa sivun tekstialueen korkeuteen. Oletusarvo on 0.3 (30 %). Kahden palstan tilassa leijuvia osia ei koskaan sijoiteta sivun alaosaan.

`\textfraction` Leipätekstin vähimmäistila sivulla, kun samalla sivulla on myös leijuvia osia. Oletusarvo on 0.2 (20 %).

`\floatpagefraction` Omalle sivulle sijoitettavien leijuvien osien vähimmäistila sivun tekstialueen korkeudesta. Toisin sanoen leijuvien osien täytyy viedä vähintään tämän verran tilaa sivulta, jotta ne voidaan sijoittaa omalle sivulle, jossa ei ole leipätekstiä lainkaan. Oletusarvo on 0.5 (50 %). Kahden palstan tilassa käytetään komentoa `\dblfloatpagefraction`.

Sivunvaihtokomennot `\clearpage` ja `\cleardoublepage` (luku 4.3) pakottavat kaikki aiemmin lähdetiedostossa olleet leijuvat osat ladottavaksi. Osanvaihtokomento `\part` ja otsikkokomento `\chapter` (luku 4.4.2) tekevät saman automaattisesti joissakin dokumenttiluokissa.

Paketin `placeins`²⁸ avulla voi tehdä omiakin rajakohtia, jonka yli leijuvat osat eivät voi siirtyä. Paketin komennolla `\FloatBarrier` tehdään ra-

²⁸ <https://www.ctan.org/pkg/placeins>

ja kyseiseen kohtaan lähdedokumentissa. Jos paketin lataamisessa antaa argumentin `section`, asetetaan tällainen raja automaattisesti `\section`-otsikoille.

```
\usepackage[section]{placeins}
```

4.9.5 Ulkoasu

Leijuvan osan sisäisen asettelun, ulkoasun ja muun typografian hallintaan ja muokkaamiseen tarvitaan yleensä paketteja `floatrow`²⁹ ja `caption`³⁰. Jälkimmäisen avulla muokataan kuvatekstin fonttia ja rivitystä, ja sitä käsitellään myöhemmin tässä alaluvussa.

Paketti `floatrow` tarjoaa runsaasti mahdollisuuksia leijuvien osien sijoitteluun: osat voi kehystää laatikolla, kuvatekstin voi sijoittaa ylös, alas tai sivulle, useita leijuvia osia voi sijoittaa vierekkäin ym. Tässä oppaassa käsitellään ainoastaan peruskäyttöä.

Jotta `floatrow`-paketin ominaisuudet saisi käyttöönsä, täytyy leijuivat osat toteuttaa sen oman `\floatbox`-komennon avulla. Komento sijoitetaan leijuvan ympäristön sisään. Sen argumenteiksi annetaan ainakin leijuvan osan tyyppi, kuvatekstikomento ja varsinainen leijuva sisältö. Seuraavassa esimerkissä käytetään `table`-ympäristöä, mutta muut tyytit toimivat samalla tavalla.

```
\begin{table}
  \floatbox[table][leveys]{kuvatekstit}{sisältö}
\end{table}
```

Valinnaisen `leveys`-argumentin voi jättää poisikin, jolloin leijuva osa varaa itselleen sivun (tai palstan) leveyden verran tilaa. Argumentiksi on joskus hyödyllistä antaa mitta `\FBwidth`, joka on leijuvan osan sisällön levyinen. Käyttämällä tätä mitta kuvatekstit rivitetään sisällön levyiseksi. Esimerkki 4.24 selventää, miten `\floatbox`-komento toimii käytännössä.

Yksi leijuva ympäristö voi sisältää rinnakkain useampia taulukoita, kuvia tms., joilla jokaisella on oma kuvatekstinsä. Se tehdään ympäristöllä `floatrow`, jonka valinnainen argumentti ilmaisee, kuinka monta

²⁹ <https://www.ctan.org/pkg/floatrow>

³⁰ <https://www.ctan.org/pkg/caption>

```

1 \begin{table}
2   \floatbox{table}[\FBwidth]{
3     \caption{Upea taulukko} % kuvateksti
4     \label{tlk/upea-tilukko} % tunniste ristiviittausta varten
5   }{
6     \begin{tabular}{ll}
7       jotain & soluja \\
8       esimerkin & vuoksi \\
9     \end{tabular}
10  }
11 \end{table}

```

Esimerkki 4.24: `\floatbox`-komennon peruskäyttö.

```

1 \begin{table}
2   \begin{floatrow}[2] % kaksi rinnakkain
3     \floatbox{table}[\FBwidth]{\caption{Vasen}}{...}
4     \floatbox{table}[\FBwidth]{\caption{Oikea}}{...}
5   \end{floatrow}
6 \end{table}

```

Esimerkki 4.25: Rinnakkaisten leijuvien osien toteutus `floatrow`-ympäristön ja `\floatbox`-komennon avulla. Kohdassa ”...” olisi varsinainen sisältö eli taulukon toteutus.

kuva tms. ladotaan rinnakkain. Oletusarvo on 2. Ympäristön sisällä täytyy olla yhtä monta `\floatbox`-komentoa. Esimerkistä 4.25 selviää perusajatus kahden taulukon latomiseksi rinnakkain.

Kun leijuvat osat on toteutettu `\floatbox`-komennon avulla, voi niiden ulkoasuun vaikuttaa komennolla `\floatsetup`. Sille annetaan ainakin yksi argumentti, joka voi sisältää useita pilkulla toisistaan erotettuja valitsimia ja niiden arvoja.

```
\floatsetup[tyyppi]{valitsimet}
```

Valinnainen argumentti tyyppi on leijuvan osan tyyppi, esimerkiksi `table` tai `figure`. Jos se on annettu, asetukset vaikuttavat vain kyseiseen tyyppiin; muuten asetukset koskevat kaikkentyyppisiä leijuvia osia. Asetukset tehdään erilaisten valitsimien avulla, joista muutama on koottu taulukkoon 4.9.

Valitsin	Merkitys ja vaihtoehtoja
style	tyyli: plain, ruled, boxed ym.
capposition	kuvatekstin sijainti: top, bottom ym.
captionskip	kuvatekstin etäisyys sisällöstä (mitta)
font	fonttiasetukset: rm, sf, tt, md, bf, up, it, sl, sc, scriptsize, footnotesize, small, normalsize, large, Large
margins	leijuvan osan sijainti: centering, raggedright, raggedleft ym.
justification	sisällön tasaus: justified, centering, raggedright, raggedleft, RaggedRight (<i>raggedze</i>), RaggedLeft (<i>raggedze</i>)

Taulukko 4.9: Muutama `\floatsetup`-komennon valitsin, joilla vaikutetaan leijuvien osien ulkoasuun. Osa asetuksista tarvitsee paketin *raggedze*.

Valitsin `font` asettaa leijuvan ympäristön fontin mutta ei vaikuta kuvatekstiin. Fonttiasetukset ovat käytännössä avainsanoja, jotka vastaavat lähes samannimisiä fonttikomentoja. Niitä käsitellään luvussa 3.4.3. Omiakin fonttien avainsanoja voi luoda komennolla `\DeclareFloatFont`. Valitsimen `justification` arvoksi annetaan myös tietty sana, joka vastaava samannimisiä palstan tasauskomentoja (luku 4.1.1). Käytännössä leijuvan osan asetukset voisivat näyttää vaikka seuraavanlaiselta:

```
\floatsetup{ style=plain, capposition=bottom, font={sf, small},
  justification=centering, captionskip=2ex }
\floatsetup[figure]{ style=boxed }
```

Kuvatekstien fonttia ja rivittämistä hallitaan `caption`³¹-paketin avulla. Tärkein komento on `\captionsetup`, joka yleensä sijoitetaan dokumentin esittelyosaan tai muuten alkuun, koska silloin se vaikuttaa koko dokumentissa. Komennon voi sijoittaa myös yksittäisen leijuvan ympäristön sisään, jolloin se vaikuttaa vain kyseisessä ympäristössä.

```
\captionsetup[tyyppi]{valitsimet}
```

Komennon valinnainen `tyyppi`-argumentti on leijuvan osan tyyppi, esimerkiksi `table` tai `figure`, johon halutaan vaikuttaa. Jos sen jättää pois, asetukset vaikuttavat kaikkien leijuvien osien kuvateksteihin. Pakollinen argumentti on `valitsimet`, johon kirjoitetaan valitsimia ja niiden

³¹ <https://www.ctan.org/pkg/caption>

Valitsin	Merkitys ja vaihtoehtoja
font	kuvatekstin fontti: normalfont, rm, sf, tt, md, bf, up, it, sl, sc, scriptsize, footnotesize, small, normalsize, large, Large
labelfont	leijuvan osan tyyppin ja numeroinnin fontti
textfont	varsinaisen kuvatekstin fontti
format	kappaleen tyyppi, tavallinen plain vai riippuva hang
indentation	sisennysmitta kuvatekstin toisesta rivistä alkaen
labelformat	leijuvan osan tyyppin muoto: default, empty, simple, brace, parens
labelsep	leijuvan osan tyyppin erotinmerkki: none, colon, period, space, quad, newline, endash
textformat	kuvatekstin lopetusmerkki: simple, period
justification	kuvatekstin tasaus: justified, centering, centerlast, centerfirst, raggedright, RaggedRight (<i>raggedze</i>), raggedleft
singlelinecheck	yksirivisten kuvatekstien poikkeuksellinen tasaus: true (oletus), false
margin	kuvatekstin marginaalit: yksi mitta tai kaksi pilkulla erotettua mitta (vasen ja oikea marginaali)
width	kuvatekstikappaleen leveys (mitta)

Taulukko 4.10: Tärkeimpiä `\captionsetup`-komennon valitsimia.

arvoja. Tärkeimpiä valitsimia on koottu taulukkoon 4.10. Käytännössä komento voisi näyttää esimerkiksi seuraavanlaiselta:

```
\captionsetup{ font={small, sf}, labelfont={bf}, textfont={},
textformat=period, margin={.5em,0em}, justification=raggedright,
singlelinecheck=false }
```

Oletuksena Latex käsittelee yksiriviset kuvatestit poikkeuksellisella tavalla eli keskittää ne. Sen vuoksi tasausvalitsin `justification` ei vaikuta yksirivisiin kuvateksteihin. Tämän poikkeuksen saa pois päältä käyttämällä asetusta `singlelinecheck=false`, jolloin sama `justification` pätee yhtä lailla niin yksi- kuin monirivisiinkin kuvateksteihin.

Joskus halutaan, että useampi taulukko, kuva tai muu leijuva osa muodostaa kokonaisuuden, jolla on sama numero, esimerkiksi ”Taulukko 3”. Ajatuksena on, että sama sisältö on jakautunut useampaan osaan. Se onnistuu sijoittamalla `caption`-paketin komento `\ContinuedFloat` ensimmäisen jälkeisiin leijuviin ympäristöihin. Komento pitäisi sijoittaa heti ympäristön alkuun. Esimerkki 4.26 havainnollistaa sen käyttöä.


```

1 \begin{table} % kokonaisuuden 1. leijuva ympäristö
2 \caption{Hieno taulukko}
3 ...
4 \end{table}
5 ...
6 \begin{table} % kokonaisuuden 2. leijuva ympäristö
7 \ContinuedFloat % sama numerointi kuin edelliselle
8 \caption{Hieno taulukko (jatkuu)}
9 ...
10 \end{table}

```

Esimerkki 4.26: Usealle leijuvalle osalle saa saman numeron käyttämällä jälkimmäisissä ympäristöissä komentoa `\ContinuedFloat`.

Kuvatekstissä oleva leijuvan osan numero tulee laskurista, ja erityyppisellä osilla on oma laskurinsa. Laskurin nimi on sama kuin vastaavan leijuvan ympäristön nimi, eli esimerkiksi `table`-ympäristöt numeroidaan laskurin `table` avulla.

Käytännössä laskurien arvo ladotaan `\the`-alkuisella komennolla, jonka perässä on laskurin nimi, esimerkiksi `\thetable`. Oletuksena nämä komennot latovat laskurin arabialaisilla numeroilla. Jos laskuri on riippuvainen teoksen pääluvuista (`\chapter`), ladotaan ensin pääluvun numero, erotinpiste ja leijuvan osan numero: 2.1, 2.2 jne.

Kirjoittaja voi määrittellä laskurien latomiskomennon haluamallaan tavalla. Seuraavassa esimerkissä leijuville kuville (`figure`) määritellään numerointitapa, jossa on pääluvun numero, ajatusviiva (-) ja leijuvan kuvan numero:

```
\renewcommand{\thefigure}{\thechapter--\arabic{figure}}
```

4.10 Ristiviittaukset

Ristiviittaukset tarkoittavat dokumentin sisäisiä viittauksia eli viittauksia toisiin kohtiin samassa dokumentissa. Lukijalle ne ilmenevät ilmauksina kuten ”luvussa 4.2” tai ”kuvassa 5”. Kirjoittajan ei kuitenkaan kannata naputella luvun numeroa (esim. 4.2) käsin lähdedokumenttiin, koska lukujen järjestys ja numerointi voi muuttua kirjoittamisen ede-

nessä. Joutuisi korjaamaan muuttuneet numerot mahdollisesti useitakin kertoja ennen kuin työ on valmis.

Latex numeroi dokumentin otsikot (luvut) ja leijuvat osat automaattisesti, eli se tietää, minkä numeron mikäkin dokumentin osa saa. Niinpä Latex osaa myös – kirjoittajan pienellä avustuksella – ylläpitää ajantasaisia ristiviittauksia dokumentin eri osiin. Teknisesti tämä toteutetaan siten, että otsikoille ja leijuville osille annetaan yksilöllinen tunniste ja viittaamisessa käytetään samoja tunnisteita. Tunnisteet annetaan `\label`-komennolla, joka sijoitetaan heti otsikkokomennon (`\section`, `\subsection` ym.) tai leijuvan osan kuvatekstikomennon (`\caption`) jälkeen:

```
\section{Jokin otsikko}
\label{tunniste}
```

`\label`-komennon argumentti tunniste voi olla suunnilleen mitä tahansa tekstiä. Sen täytyy olla yksilöllinen, eli sama tunniste ei saa olla käytössä missään toisessa `\label`-komennossa. Jos sama tunniste sattuu olemaan useassa paikassa, vain viimeinen jää käytännössä voimaan.

Järkevää on kirjoittaa tunnisteeseen sana tai sanoja, jotka kertovat jotakin kyseisen osan sisällöstä. Hyödyllistä voi olla myös liittää mukaan tieto, onko kyse luvusta, taulukosta, kuvasta vai muusta leijuvasta osasta, koska erityyppiset osat voivat käsitellä samaa aihetta. Seuraavassa on esimerkkejä `\label`-komennoista:

```
\label{luku/fonttien-valinta} % otsikkokomennon jälkeen
\label{tlk/fonttikomentoja} % taulukon \caption-komennon jälkeen
\label{kuva/kirjainleikkauksia} % kuvan \caption-komennon jälkeen
```

Ristiviittauksissa vähimmällä vaivalla selviää, kun huomioi kaksi asiaa: 1) kirjoittaa `\label`-komentoja vain niihin kohtiin, joihin täytyy viitata muualta, 2) yrittää valita tunnisteet siten, ettei niitä tarvitse muuttaa enää sen jälkeen, kun ne on kerran valinnut.

Tavallisimmat ristiviittaukset tehdään `\ref`-komennolla, jonka argumentiksi annetaan sama tunniste kuin jossakin dokumentin `\label`-komennossa. Latex latoo `\ref`-komentojen kohdalle kyseisen luvun tai leijuvan osan numeron. Toinen tarpeellinen ristiviittauskomento on `\pageref`, joka kääntämisvaiheessa korvautuu kyseisen kohteen

sivunumerolla. Näitä voi hyödyntää lähdedokumentissa esimerkiksi seuraavasti:

```
Katso taulukko \ref{tlk/fonttimallikomennot} sivulla
\pageref{tlk/fonttimallikomennot}.
```

⇒ Katso taulukko 3.8 sivulla 84.

Jos ladattuna on paketti `hyperref` (luku 3.3), on ristiviittauksiin käytävissä myös komento `\nameref`. Sillekin annetaan argumentiksi jokin `\label`-komennolla määritetty tunniste, ja kääntämisvaiheessa komennon tilalle ladotaan kyseisen kohteen teksti eli otsikko tai kuvateksti. Ristiviittauksissa ladotaan otsikon tai kuvatekstin lyhempi versio, jos sellainen on annettu. Otsikkokomentojen ja kuvatekstikomennon argumentteja käsitellään luvuissa 4.4.2 (Otsikointi) ja 4.9.1 (Leijuvat taulukot ja kuvat).

Ristiviittaukset vaativat, että Latex-lähdedokumentti käännetään kahdesti. Ensimmäisellä kääntämiskerralla `\label`-komennolla mainittujen kohteiden tiedot kirjoitetaan muistiin väliaikaistiedostoon. Toisella kääntämiskerralla hyödynnetään väliaikaistiedostoa ja `\ref`-, `\pageref`- ja `\nameref`-komentojen tilalle ladotaan viittauskohteen oikeat tiedot.

4.11 Alaviitteet

Alaviitteet ovat sivun alareunassa olevia numeroituja tai muulla tavalla merkittyjä huomautuksia. Niihin viitataan sanojen tai virkkeiden perässä olevalla yläindeksinumerolla, -kirjaimella tai -symbolilla.³² Alaviitteiden tarkoituksena on lisätä tekstiin lähes huomaamaton lisätieto, joka ei häiritse lukemista. Se sopii asioille, joilla ei ole suurta merkitystä useimmille lukijoille tai jotka muuten sopivat huonosti leipätekstiin. Varsin usein myös tiedonlähteet ilmaistaan alaviitteiden avulla.³³

Alaviite tehdään komennolla `\footnote`, jonka argumentiksi kirjoitetaan alaviitteeseen tuleva teksti. Oletuksena Latex latoo komennon

³² Tässä on esimerkki alaviitteestä.

³³ Kankaanpää ym. 2017, s. 162; Itkonen 2019, s. 127–128.

paikalle yläindeksinumeron, joka tulee automaattisesti laskurista `footnote`. Komennolle voi antaa myös valinnaisen argumentin, joka on kyseisen alaviitteen numero. Tässä tapauksessa komento ei kasvata `footnote`-laskuria.

```
\footnote{Alaviitteen teksti.}
\footnote[numero]{Alaviitteen teksti.}
```

Itse alaviite ladotaan samalla kirjaintyyppillä kuin leipätekstikin mutta selvästi pienempänä: sen koko tulee komennosta `\footnotesize` (luku 3.4.3). Alaviitteet erotetaan leipätekstistä pienellä pystysuuntaisella välillä sekä oletuksena myös lyhyellä vaakaviivalla. Samalle sivulle voi sattu useampikin alaviite. Jos alaviitteen tekee leijuvaan osaan (luku 4.9) tai `minipage`-ympäristöön (luku 2.6.2), se merkitään oletuksena kirjaimen avulla, ja alaviitteen teksti ladotaan kyseisen osan alapuolelle.

`\footnote`-komennon toiminnan voi jakaa kahteen erilliseen osaan komentojen `\footnotemark` ja `\footnotetext` avulla. Ensin mainittu komento latoo pelkän yläindeksinumeron, joka tulee alaviitelaskurista. Komento ei lado varsinaista alaviitettä. Jälkimmäinen komento puolestaan latoo pelkän alaviitteen. Kummallekin komennolle voi antaa valinnaisen argumentin, jolla ilmaistaan kyseisen alaviitteen numero.

```
\footnotemark[numero] % yläindeksinumero
\footnotetext[numero]{Alaviitteen teksti.} % alaviite
```

Alaviitteiden numerointi on laskurissa `footnote`, mutta sen arvo ladotaan komennon `\thefootnote` avulla. Tämän komennon määritelmässä on oletuksena komento `\arabic{footnote}`, eli laskuri ladotaan arabialaisilla numeroilla. Kirjoittaja voi kuitenkin määritellä komennon uudelleen. Seuraavassa esimerkissä määritellään komento `\siten`, että alaviitteet ilmaistaan symbolien avulla:

```
\renewcommand{\thefootnote}{\fnsymbol{footnote}}
```

Laskurien symboliesitys toimii vain lukualueella 1–9, joten samaan dokumenttiin ei mahdu kovin monta alaviitettä, ellei laskuria nollaa välillä. Laskurin voi määritellä riippuvaiseksi toisesta laskurista, jolloin se

nollautuu itsestään, kun toisen laskurin arvo kasvaa. Esimerkiksi dokumenttiluokassa `book` alaviitteiden laskuri nollautuu oletuksena aina päälukujen (`\chapter`) vaihtuessa.

Joskus voi olla tarpeen tehdä alaviitteiden laskuri sivukohtaiseksi eli riippuvaiseksi `page`-laskurista. Sen voi toteuttaa esimerkiksi paketin `footmisc` ja valitsimen `perpage` avulla. Näitä käsitellään myöhemmin tässä luvussa. Toinen vaihtoehto sivukohtaiseen alaviitenumerointiin on `chngcntr`-paketin komento `\counterwithin*` (luku 2.5.1). Seuraavassa on siitä esimerkki:

```
\counterwithin*{footnote}{page}
```

Edellisissä esimerkeissä on käsitelty koko dokumentin alaviitelaskuria `footnote`, mutta leijuissa osissa ja `minipage`-ympäristöissä (luku 2.6.2) alaviitteet numeroidaan laskurin `mpfootnote` avulla. Tämän laskurin arvo ladotaan käytännössä komennolla `\thempfootnote`, jonka kirjoittaja voi määrittellä uudelleen tarpeen mukaan.

Leipäteksti ja alaviitteet erotetaan pystysuuntaisella välillä, jonka voi asettaa sivun asetusten yhteydessä (luku 3.2). Asetus on `geometry`-paketin valitsin `footnotesep`, jota käytetään esimerkiksi `geometry`-komenton kanssa:

```
\geometry{footnotesep=14bp}
```

Välin täytyy olla ainakin niin suuri, ettei lukija sekoita alaviitteitä leipätekstin kappaleeseen. Riittävä väli on noin yhden rivikorkeuden verran tai vähän enemmän. Sopivan välin suuruus voi riippua siitäkin, erotetaanko leipäteksti ja alaviitteet toisistaan myös vaakaviivalla.

Muunlaiset alaviitteisiin vaikuttavat asetukset täytyy tehdä paketin `footmisc`³⁴ avustuksella. Sen avulla voi esimerkiksi poistaa alaviitteiden erotinviivan ja muotoilla alaviitekappaleita. Paketin lataamisen yhteydessä voi antaa useita pilkulla erotettuja valitsimia, joita on koottu taulukkoon 4.11. Erotinviiva poistetaan paketin valitsimella `norule`. Samalla pystysuuntainen väli hieman kasvaa.

```
\usepackage[norule]{footmisc}
```

³⁴ <https://www.ctan.org/pkg/footmisc>

Valitsin	Merkitys
<code>norule</code>	vaakasuintainen erotinviiva pois
<code>perpage</code>	numerointi sivukohtaiseksi
<code>bottom</code>	alaviite aina sivun alareunaan
<code>hang</code>	luetelmatyypiset alaviitteet, riippuva sisennys
<code>multiple</code>	useat peräkkäiset yläindeksinumerot

Taulukko 4.11: `footmisc`-paketin asetusten avulla muotoillaan alaviitteitä.

Valitsin `perpage` numeroi alaviitteet sivukohtaisesti, eli niiden laskuri nollautuu aina sivun vaihtuessa. Valitsin `bottom` pakottaa alaviitteen aina sivun alareunaan. Oletusasetuksilla alaviite voi sijoittua joskus ylemmäksi, jos sivun alaosaan ladotaan leijuva osa tai jos sivun alareunan tasaaminen on pois päältä (`\raggedbottom`, luku 4.3).

Oletuksena alaviitekappaleet ladotaan samalla tavalla kuin tekstikappaleet oletuksena muutenkin, eli alaviitteen ensimmäinen rivi on sisennetty, ja sen alussa on alaviitteen numero. Seuraavia rivejä ei sisennetä. Tällainen asetus sopii pitkien alaviitetekstien latomiseen. Usein viitteet ovat kuitenkin lyhyitä, ja silloin voi sopia paremmin luetelmatyypiset alaviitekappaleet (riippuva sisennys), joissa alaviitteen numero on omalla alueellaan ja varsinainen alaviiteteksti on sisennetty. Tällainen asetus tehdään `footmisc`-paketin valitsimella `hang`. Sisennyksen suuruuden voi vaikuttaa mitan `\footnotemargin` avulla:

```
\usepackage[hang]{footmisc}
\setlength{\footnotemargin}{1.5em}
```

Valitsin `multiple` mahdollistaa peräkkäisten `\footnote`-komentojen kirjoittamisen helposti. Peräkkäin olevien yläindeksinumeroitten väliin ladotaan automaattisesti pilkku, kuten seuraava esimerkki osoittaa:

```
sana\footnote{...}\footnote{...}
⇒ sana7,8
```

Toisaalta yläindeksiin voi sijoittaa pilkun tai muuta tekstiä käsinkin komennolla `\textsuperscript` (luku 2.1.12). `multiple`-valitsin vaatii toimiakseen tietyn `hyperref`-paketin valitsimen, joka käsitellään seuraavaksi.

Jos ladattuna on `hyperref`-paketti (luku 3.3), alaviitteisiin liittyvistä yläindeksinumeroista tulee myös hiirellä napsautettavia linkkejä. Tätä ominaisuudesta on harvoin mitään hyötyä, koska alaviite on melkein aina samalla sivulla. Linkkitoiminto myös häiritsee `footmisc`-paketin `multiple`-valitsimen toimintaa. Alaviitteiden linkit voi kytkeä pois `hyperref`-paketin lataamisen yhteydessä seuraavalla asetuksella:

```
\usepackage[hyperfootnotes=false]{hyperref}
```

`footmisc`-paketti sisältää muitakin toimintoja alaviitteiden ulkoasun säätämiseen. Lisätietoa saa paketin ohjekirjasta.

4.12 Palstat

Latex pystyy usean palstan käsittelyn perusasioihin, mutta mihinkään sanoma- tai aikakauslehden kaltaiseen monipuoliseen tekstiin, kuvien ja muiden osien sommitteluun se ei kykene. Verrattain yksinkertaiset monipalstaiset tekstit kuitenkin onnistuvat Latexissakin varsin mukavasti.

Perus Latexissa on kaksi palstoihin liittyvää tilaa: normaali eli yhden palstan tila sekä kahden palstan tila, joka täytyy erikseen kytkeä päälle. Tätä tilaa käsitellään luvussa 4.12.1. Toinen palstojen toteuttamisen vaihtoehto on paketti `multicol`. Sen kanssa käytetään Latexin yhden palstan tilaa ja erityistä palstaympäristöä `multicols`. Ympäristön sisällä teksti rivitetään kahdelle tai useammalle palstalle. Tätä vaihtoehtoa käsitellään luvussa 4.12.2.

4.12.1 Kahden palstan tila

Perus Latexin yhden tai kahden palstan tilan voi valita jo dokumenttiluokan (luku 3.1.2) lataamisen yhteydessä tai sivun asetuksissa (luku 3.2) käyttämällä valitsinta `onecolumn` (oletus) tai `twocolumn`. Dokumentin keskellä tilaa voi vaihtaa komennoilla `\onecolumn` ja `\twocolumn`. Kumpikin komento aloittaa uuden sivun, josta alkaen valittu tila on voimassa. Komennolle `\twocolumn` voi antaa hakasulkeissa valinnaisen argumentin, jolla ilmaistaan sivun alkuun ladottava koko sivun levyinen sisältö:

```
\twocolumn[yksipalstaista sisältöä]
```

`\twocolumn`-komennon valinnainen argumentti mahdollistaa sen, että sivun alkuun voi sijoittaa esimerkiksi koko sivun levyisen otsikon, kuvan tai muun osan. Sen jälkeen lähes kaikki sisältö on ladotaan kahdelle palstalle, eli teksti täyttää ensin vasemmanpuoleisen palstan ja sen jälkeen oikeanpuoleisen.

Leijuvat osat (luku 4.9) ja ympäristöt kuten `table` ja `figure` sijoitetaan palstan sisään, eli niiden sisältö täytyy suunnitella yhden palstan levyiseksi. Ympäristöjen tähdelliset versiot kuten `table*` ja `figure*` ladotaan koko sivun levyisenä sivun yläosaan tai kokonaan omalle sivulleen, jossa on vain leijuvia osia. Leijuvien osien sijoitteluun kahden palstan tilassa voi vaikuttaa tiettyjen laskurien, mittojen ja kommentojen avulla. Niitä käsitellään luvussa 4.9.4.

Palstojen välissä olevan tyhjän tilan leveys on mitassa `\columnsep`, ja sen voi asettaa haluamakseen `\setlength`-komennolla tai `geometry`-paketin asetuksista (luku 3.2.1). Oletusasetus on 35 pt. Palstojen väliin saa myös pystyviivan, jonka leveys on mitassa `\columnseprule`. Sen oletusarvo on nolla, eli pystyviivaa ei ladota.

```
\setlength{\columnsep}{6mm}      % palstojen väli
\setlength{\columnseprule}{.2mm} % pystyviivan leveys
```

Latex laskee automaattisesti palstan leveyden ja asettaa sen mittaan `\columnwidth`. Tätä mittaa ei siis tarvitse eikä pidä itse asettaa, mutta sitä voi hyödyntää esimerkiksi taulukoiden tai kuvien sovittamisessa koko palstan levyiseksi. Käytännössä kahden palstan tilassa palstan leveys on mittojen `\textwidth` ja `\columnsep` erotus jaettuna kahdella. Tilannekohtainen rivin pituus on mitassa `\linewidth`.

4.12.2 Palstaympäristö

Perus Latex osaa vain yhden ja kahden palstan tilan, ja ne ovat voimassa koko sivulla. Paketti `multicol`³⁵ tuo käyttöön myös ympäristön `multicols`, jonka sisällä tietty palstamäärä on voimassa. Samallakin sivulla voi olla eri tavoin palstoitettua sisältöä. Esimerkki 4.27 havainnollistaa ympäristön käyttöä ja ladottua lopputulosta. Oleelliset tiedot ovat seuraavat:

³⁵ <https://www.ctan.org/pkg/multicol>


```

1 \begin{multicols}{3} % kolme palstaa
2 \noindent Jukolan talo, eteläisessä Hämeessä, seisoo erään mäen
3 pohjoisella rinteellä, liki Toukolan kylää. Sen läheisin ympäristö on
4 kivinen tanner, mutta alempana alkaa pellot, joissa, ennenkuin talo
5 oli häviöön mennyt, aaltoili teräinen vilja. Peltojen alla on niittu,
6 apilaäyräinen, halkileikkaama monipolvisen ojan; ja runsaasti antoi se
7 heiniä, ennenkuin joutui laitumeksi kylän karjalle. Muutoin on talolla
8 avaria metsiä, soita ja erämaita, jotka, tämän tilustan ensimmäisen
9 perustajan oivallisen toiminnan kautta, olivat langenneet sille osaksi
10 jo ison jaon käydessä entisinä aikoina.
11 \end{multicols}

```

⇒ Jukolan talo, eteläisessä Hämeessä, seisoo erään mäen pohjoisella rinteellä, liki Toukolan kylää. Sen läheisin ympäristö on kivinen tanner, mutta alempana alkaa pellot, joissa, ennenkuin talo oli häviöön mennyt, aaltoili teräinen vilja. Peltojen alla on niittu, apilaäyräinen, halkileikkaama monipolvisen ojan; ja runsaasti antoi se heiniä, ennenkuin joutui laitumeksi kylän karjalle. Muutoin on talolla avaria metsiä, soita ja erämaita, jotka, tämän tilustan ensimmäisen perustajan oivallisen toiminnan kautta, olivat langenneet sille osaksi jo ison jaon käydessä entisinä aikoina.

Esimerkki 4.27: Palstojen toteutus `multicol`-paketin `multicols`-ympäristön avulla. Teksti on Aleksis Kiven *Seitsemän veljestä* -romaanin alusta.

```

\begin{multicols}{palstamäärä}
...
\end{multicols}

```

Argumentti `palstamäärä` on kokonaisluku, joka ilmaisee, kuinka monelle palstalle ympäristön sisältö ladotaan. Palstojen korkeus pyritään tasaamaan automaattisesti, niin että kaikki palstat ovat yhtä korkeita. Käytännössä palstaympäristön viimeisen sivun viimeinen palsta jää hieman vajaaksi. Jos käyttää ympäristön tähtiversiota `multicols*`, ei viimeisen sivun palstojen korkeuksia tasata, vaan ensin täytetään vasemmanpuoleinen palsta kokonaan, sitten seuraava jne.

Ympäristölle voi antaa hakasulkeissa valinnaisen argumentin, joka voi sisältää mitä tahansa tekstiä tai koodia. Se on tarkoitettu otsikolle (`\section` ym.) tai muulle sisällölle, joka ladotaan ennen palstoja ja jonka täytyy päätyä samalle sivulle kuin palstasisällön alku.

```

\begin{multicols}{palstamäärä}
  [tekstiä tai koodia ennen palstoja]
  ...
\end{multicols}

```

Ennen palstaympäristöä ja sen jälkeen ladotaan mitan `\multicolsep` suuruinen pystysuuntainen väli. Väliä ei kuitenkaan ladota, jos se sattuu sivunvaihdon kohdalle. Palstojen välissä oleva vaakasuuntainen tyhjä tila asetetaan mitan `\columnsep` avulla, ja väliin mahdollisesti ladottavan pystyviivan leveys on mitassa `\columnseprule`. Tämän mitan oletusarvo on nolla, eli pystyviivaa ei ladota.

```

\setlength{\multicolsep}{.5ex plus .2ex} % väli ennen ja jälkeen
\setlength{\columnsep}{6mm}             % palstojen väli
\setlength{\columnseprule}{.2mm}        % pystyviivan leveys

```

Oletuksena palstojen alareuna pyritään joka sivulla pitämään tarkasti samalla tasolla. Tämä toteutuu siten, että palstoilla olevia pystysuuntaisia välejä voidaan venyttää, jotta niiden viimeiset rivit saataisiin samalle tasolle. Tämä on yhdenmukainen sivujen tasaamiskäytännön kanssa, jota käsitellään luvussa 4.3. Palstojen alareunan tasaamisen ja pystysuuntaisten välien venyttämisen saa pois komennolla `\raggedcolumns`. Oletustilaan palataan komennolla `\flushcolumns`. Palstanvaihdon voi pakottaa tiettyyn kohtaan komennolla `\columnbreak`. Jos komento sijaitsee tekstikappaleen sisällä, palsta vaihtuu nykyisen rivin lopussa.

Leijuvat osat (luku 4.9) toimivat `multicols`-ympäristössä hieman toisin kuin muualla. Leijuvien ympäristöjen tähdellisiä versioita kuten `table*` ja `figure*` ei koskaan ladota samalle sivulle kuin missä ne esiintyvät. Ensimmäinen mahdollinen paikka on seuraavan sivun yläosassa.

Leijuvien osien sijoitteluun vaikuttavat tavalliset yhden palstan tilan asetukset kuten laskuri `topnumber`, mitta `\floatsep` ja komento `\topfraction`. `multicols`-ympäristössä ei käytetä `dbl`-alkuisia kahden palstan tilan laskureita, mittoja eikä komentoja, kuten `dbltopnumber`, `\dblfloatsep` ja `\dbltopfraction`. Lisätietoa leijuvien osien sijoittelusta on luvussa 4.9.4.

4.13 Lähdeluettelo ja lähdeviitteet

Lähdeluettelossa mainittuihin teoksiin on tapana viitata muualta tekstistä, ja Latex osaa auttaa viittausten hallinnassa. Ajatus on se, että lähdeluettelo laaditaan tiettyjen kommentojen avulla niin, että jokainen lähde-teos saa yksilöllisen tunnisteiden. Muualta tekstistä viitataan lähde-teoksiin käyttämällä samoja tunnisteita, ja Latex osaa automaattisesti poimia lähdeluettelosta esimerkiksi teoksen tekijöiden nimet ja vuosiluvun.

Lähdeviittauksiin ja lähdemerkintöihin on useita käytäntöjä, jotka vaihtelevat eri ammatti- ja tieteenaloilla, oppilaitoksilla tai julkaisijoilla. Tässä yhteydessä käsitellään melko vakiintuneita suomalaisia käytäntöjä, jotka kuvataan *Kielitoimiston oikeinkirjoitusoppaassa* (Kankaanpää ym. 2017). Samalla käsitellään joitakin asetuksia, joilla kukin voi muokata viittausten ulkoasua omiin tarpeisiin sopivaksi.

Latex sisältää lähdeluetteloiden ja -viittausten perustoiminnot, mutta niiden avulla ei saa yleisen suomalaisen käytännön mukaisia lähdeviittauksia. Siksi usein tarvitaan avuksi paketti, jolla viittausten ja lähdeluettelon ulkoasuun voi vaikuttaa. Ensin käsiteltävä paketti `natbib` (luku 4.13.1) soveltuu perustarpeisiin ja lienee sopivin valinta useimmille kirjoittajille. Laajoja tieteellisiä teoksia kirjoittavan kannattanee opetella käyttämään monipuolista `biblatex`-pakettia (luku 4.13.2) ja ylläpitää yhteistä lähde-teosten tietokantaa, josta tarvittavat teokset poimitaan kunkin dokumentin lähdeluetteloon automaattisesti.

4.13.1 Peruskäyttöön (`natbib`)

Paketti `natbib`³⁶ laajentaa Latexin lähdeviittausten perustoimintoja sen verran, että lähteisiin voidaan viitata teoksen tekijöiden nimen ja vuosiluvun avulla. Seuraava esimerkki havainnollistaa paketin käyttöön-ottoa ja asetuksia.

```
\usepackage{natbib}
\setcitestyle{authoryear,aysep={},notesep={:}}
```

³⁶ <https://www.ctan.org/pkg/natbib>

Edellisessä esimerkissä viittaustyyli valitaan `\setcitetstyle`-komennon argumentissa valitsimella `authoryear` (tekijä–vuosi). Valitsimella `aysep` määritetään, mikä välimerkki ladotaan tekijän nimen ja vuosiluvun väliin. Tässä se jätetään tyhjäksi. `notesep`-valitsimella asetetaan merkit, jotka ladotaan vuosiluvun ja sitä seuraavan huomautuksen kuten sivunumeroiden väliin; tässä tapauksessa määritettiin kaksoispiste ja väli `{: }`, mutta pilkkukin on yleinen käytäntö.

`\setcitetstyle`-komennon valitsimet erotetaan toisistaan pilkulla, eikä erotinpilkkujen ympärillä saa olla välilyöntejä. Lopputuloksena lähdemerkinnät näyttävät esimerkiksi seuraavanlaisilta:

```
\citet*[27--29]{johdatus} % Viittaus teokseen "johdatus".
```

⇒ Meikäläinen & Teikäläinen (2020: 27–29)

Lähdeluettelo kirjoitetaan `thebibliography`-ympäristön ja `\bibitem`-komentojen avulla esimerkin 4.28 tavoin. Ympäristön aloittavan komennon (rivi 1) yhteydessä on argumentti 00, jolla ei ole tässä yhteydessä merkitystä. Jos lähdeviittauksen tyylinä olisi `numbers` (eikä `authoryear`), lähdeluettelon teokset numeroitaisiin, ja silloin `thebibliography`-ympäristön argumentti ilmaisee, kuinka leveään sisennyksen numeroidut teokset tarvitsevat. Argumentiksi voi kirjoittaa mitä tahansa merkkejä, ja Latex mittaa niiden leveyden. Kannattaa kirjoittaa leveitä numeroita kuten nollia (0) niin monta kappaletta kuin on numeroita suurimmassa lähdemerkinnän luvussa. Yksi nolla riittää, jos lähteitä on 1–9 kappaletta, kaksi jos lähteitä on kaksinumeroinen määrä eli 10–99 kappaletta jne.

Komennolla `\bibitem` tehdään varsinaiset teosmerkinnät. Samalla määritetään teoksen yksilöllinen tunniste ja mitä tietoja lähdeviittauksissa näytetään. Yleinen muoto on seuraavanlainen:

```
\bibitem[lyhyt(vuosi)pitkä]{tunniste} Lähdeluettelon tekstit.
```

Valinnaisen argumentin aluksi kirjoitetaan lähdeviittauksen lyhyt merkintä, joka tulisi näkymään lähdeviittauksissa esimerkiksi muodossa ”Meikäläinen ym.”. Heti sen perään kirjoitetaan sulkeissa teoksen vuosiluku ja sen perään vapaavalintainen lähdeviittauksen pitkä merkintä,

```

1 \begin{thebibliography}{00}
2
3 \bibitem[Meikäläinen ym.(2020)Meikäläinen \& Teikäläinen]{johdatus}
4   Meikäläinen, Matti \& Teikäläinen, Teija (2020): Johdatus alkeiden
5   perusteisiin. Toinen painos. Kustantaja oy.
6
7 \bibitem[Itkonen(2019)]{typografia} Itkonen, Markus (2019): Typografian
8   käsikirja. Viides, tarkistettu painos. Typoteekki. Graafinen
9   suunnittelu Markus Itkonen oy.
10
11 \end{thebibliography}

```

Esimerkki 4.28: Lähdeluettelon kirjoittaminen `thebibliography`-ympäristön ja `\bibitem`-komentojen avulla..

Komento	Esimerkki
<code>\citet</code>	Meikäläinen ym. (2020)
<code>\citet*</code>	Meikäläinen & Teikäläinen (2020)
<code>\citep</code>	(Meikäläinen ym. 2020)
<code>\citep*</code>	(Meikäläinen & Teikäläinen 2020)
<code>\citealt</code>	Meikäläinen ym. 2020
<code>\citealt*</code>	Meikäläinen & Teikäläinen 2020
<code>\citeauthor</code>	Meikäläinen ym.
<code>\citeauthor*</code>	Meikäläinen & Teikäläinen
<code>\citeyear</code>	2020
<code>\citeyearpar</code>	(2020)

Taulukko 4.12: `natbib`-paketin lähdeviittauskomentoja.

joka näkyisi esimerkiksi tekstinä ”Meikäläinen & Teikäläinen”. Vuosiluvun sulkeiden ympärillä ei saa olla välilyöntejä.

`\bibitem`-komennon pakollinen argumentti on kyseisen lähdeteksen yksilöllinen tunniste, jonka avulla teokseen viitataan. Komennon argumenttien jälkeen kirjoitetaan samaan tekstikappaleeseen teksti, joka tulee näkymään lähdeluettelossa.

Lähdeteksiin viittaamiseen on useita eri komentoja, jotka eroavat toisistaan siinä, mitä tietoa lähdeviittauksessa näytetään ja onko lähdeviittaus tai sen osa sulkeissa vai ei. Taulukossa 4.12 on joitakin `natbib`-paketin viittauskomentoja sekä esimerkki viittauksen ulkoasusta. Kuvitelu esimerkkiteos on peräisin esimerkistä 4.28.

Lähdeluettelon ulkoasuun voi vaikuttaa mittojen `\bibhang` ja `\bibsep` avulla. Ensin mainittu on lähdemerkinnän vaakasuuntaisen riippuvuuden sisennyksen suuruus, ja jälkimmäinen on lähdemerkintöjen välinen pystysuuntainen tila.

```
\setlength{\parindent}{1.1em} % tekstikappaleiden 1. rivin sisennys
\setlength{\bibhang}{\parindent}
\setlength{\bibsep}{.5ex plus .1ex minus .1ex}
```

Lähdemerkintöjen fonttiin voi vaikuttaa määrittelemällä uudelleen komennon `\bibfont` ja sijoittamalla halutut fontti- tai muut komennot kyseisen komennon määritelmään.

```
\renewcommand{\bibfont}{\sffamily\small}
```

Oletuksena `thebibliography`-ympäristö latoo lähdeluettelolle otsikon, ja otsikon teksti määräytyy kieliasetusten (luku 3.5) ja dokumenttiluokan perusteella (luku 3.1). Suomenkielisen lähdeluettelon otsikon voi määrittää dokumentin esittelyosassa seuraavan esimerkin tavoin. Esimerkissä hyödynnetään `\addto`-komentoa, joka sisältyy `polyglossia`- ja `babel`-paketteihin.

```
\addto{\captionsfinnish}{%
  \renewcommand{\refname}{Lähteet} % article-dokumenttiluokka
  \renewcommand{\bibname}{Lähteet} % report- ja book-luokat
}
```

On myös mahdollista määritellä koko komentosarja, joka suoritetaan lähdeluettelon otsikoinnin yhteydessä. Se tehdään määrittelemällä uudelleen komento `\bibsection`.

```
\renewcommand{\bibsection}{%
  \setcounter{secnumdepth}{-1}
  \section{Lähteet}
}
```

Edellisessä esimerkissä komennolla `\setcounter` määritetään, mille otsikkotasolle dokumentin otsikoiden eli lukujen numerointi yltää. Pieni arvo (-1) käytännössä tarkoittaa, että seuraaviin otsikoihin ei tule numerointia. Komento `\section` tekee itse otsikon.

Jos ei halua, että `thebibliography`-ympäristö tekee otsikon automaattisesti, voi `\bibsection`-komennon määrittää tyhjäksi.

```
\renewcommand{\bibsection}{} 
```

Paketti `natbib` sisältää edellä kuvattujen lisäksi muitakin ominaisuuksia, joihin voi tutustua paketin ohjekirjan avulla. On muun muassa mahdollista tehdä lähdeoteoksista tietokanta Bibtex-järjestelmän avulla. Jos kuitenkin siihen suuntaan haluaa edetä, ei ehkä kannata käyttää `natbib`-pakettia eikä vanhaa Bibtexiä vaan monipuolisempaa pakettia `biblatex`, jota käsitellään seuraavassa alaluvussa.

4.13.2 Vaativaan käyttöön (`biblatex`)

Suurten lähde- ja kirjallisuusluetteloiden ylläpito voi olla aika työlästä: pitää jatkuvasti varmistaa, että kaikki viitatus teokset ovat luettelossa ja että luettelo on pilkulleen yhdenmukainen. Paketti `biblatex`³⁷ on vastaus sellaisiin tarpeisiin.

Ajatuksena on, että kaikki tiedonlähteet ja kirjallisuus kirjoitetaan tietokantaan, josta `biblatex`-paketin komennot hakevat tiedot automaattisesti. Kirjoittaja tai työryhmä voi ylläpitää yhtä kirjallisuustietokantaa, joka voi olla saatavilla oman laitoksen verkkopalvelimella tai julkisella verkkosivullakin. Dokumentin tekstissä viitataan teoksiin yksilöllisen tunnisteiden avulla, ja pelkän viittauksen perusteella oikeat teokset ilmestyvät lähdeluetteloon automaattisesti aakkosjärjestyksessä ja yhdenmukaisessa muodossa. Yhtään tiedonlähdetä ei tarvitse kirjoittaa lopulliseen lähdeluetteloon käsin.

`biblatex`-paketin käyttö vaatii hieman opettelua – varsinkin jos on tarve muokata lähdeluettelon ja lähdeviittausten ulkoasua. Muutaman tiedonlähteen ylläpito on todennäköisesti paljon helpompaa ja nopeampaa 4.13.1-paketin keinoilla (luku 4.13.1). Sen sijaan laajoja tieteellisiä artikkeleita kirjoittaville `biblatex` voi olla suuri apu, koska artikkeleissa on yleensä paljon lähteitä ja useissakin artikkeleissa viitataan yleensä samoihin lähteisiin.

TEOSTIETOKANTA

Lähdeoteosten tietokanta on erillinen tekstitiedosto, joka tavallisesti nimetään `bib`-päätteiseksi, esimerkiksi `teokset.bib`. Tiedosto koostuu

³⁷ <https://www.ctan.org/pkg/biblatex>

@-merkillä ja teostyyppin nimellä alkavista tietueista, joiden yleinen muoto on seuraavanlainen:

```
@teostyyppi{tunniste,  
  author = {...},  
  title = "..."  
}
```

Teostyyppin nimen jälkeen aaltosulkeiden sisään kirjoitetaan teoksen kaikki tiedot. Ne alkavat teoksen yksilöllisellä tunnisteella, jota käytetään lähdeviittauksissa. Tunnisteen jälkeen tulevat muut kentät. Eri kentät kuten `author` ja `title` erotetaan toisistaan pilkulla. Kentän nimi ja sen sisältö erotetaan toisistaan yhtäsuuruusmerkillä (=), ja kentän sisältö kirjoitetaan aaltosulkeiden tai lainausmerkkien sisään, kuten edellinen esimerkki näyttää.

Todellista käyttöä vastaava tietokanta tai sen osa on esimerkissä 4.29, jossa on neljä erityyppistä teostietuetta: `book`, `incollection`, `article` ja `online`. Ensin mainittu teostyyppi `book` sopii tavallisille kirjoille, joissa tietyt tekijät (`author`) vastaavat suunnilleen koko teoksen sisällöstä ja teoksella on jokin julkaisijataho (`publisher`).

Teostyyppi `incollection` tarkoittaa esimerkiksi artikkelikokoelmaa, jonka yksittäiseen artikkeliin (`title`) ja sen kirjoittajaan (`author`) on tarkoitus viitata. Voidaan mainita myös artikkelin alku- ja loppusivut (`pages`). Kokoelmalla on toimittaja (`editor`) ja yhteinen nimi (`booktitle`).

Tyyppi `article` sopii säännöllisesti julkaistavan aikakaus- tai muun lehden artikkeleihin. Viittaustarkoituksena on yksittäinen artikkeli ja sen kirjoittaja. Julkaisutiedoissa mainitaan lehden nimi (`journaltitle`), julkaisukausi (`volume`), kauteen kuuluvan julkaisun järjestysnumero (`number`) sekä mahdollisesti artikkelin sivut (`pages`).

Verkkolähteiden merkitsemiseen sopii `online`-teostyyppi, joissa on tavanomaisten kenttien lisäksi ainakin verkko-osoite eli `url`-kenttä ja mahdollisesti viittauspäivä (`urldate`) osoittamassa, milloin viitatus tiedot olivat saatavilla.

Teostyyppisiä ja teoksiin liittyviä tietokenttiä on olemassa paljon muitakin. Niiden merkitystä ja käyttöä neuvotaan tarkemmin `bibtex`-paketin ohjeissa. Seuraavassa on kuitenkin pari huomiota tietokannan ja kenttien kieliopillisista asioista.


```

1 @book{itkonen_typogr,
2   author = {Itkonen, Markus},
3   title = {Typografian käsikirja},
4   date = {2019},
5   edition = {5},
6   publisher = {Typoteekki. Graafinen suunnittelu Markus Itkonen Oy}
7 }
8
9 @incollection{likonen_teams,
10  author = {Likonen, Teemu and Riskilä, Kaisa},
11  title = {Verkkoyhteistyö Teams-ympäristössä},
12  editor = {Tammi, Tuomo and Horila, Mikko},
13  booktitle = {Oppimis- ja toimintaympäristöjen kehittäminen
14    harjoittelukouluissa II},
15  booksubtitle = {Tilat ja tekniikka pedagogisen kehittämisen tukena},
16  publisher = {E-norssi. Opettajankouluttajien yhteistyöverkosto},
17  date = {2020},
18  pages = {85-92},
19  url = {http://www.enorssi.fi/oppimisymparistojulkaisu2020/}
20 }
21
22 @article{likonen_tietokanta,
23  author = {Likonen, Teemu},
24  title = {Tietoa kantaan ja takaisin},
25  journaltitle = {Skrolli},
26  journalsubtitle = {Tietokonekulttuurin erikoislehti},
27  date = {2015},
28  volume = {2015},
29  number = {4},
30  pages = {52-55},
31  url = {https://skrolli.fi/numerot/2015-4/}
32 }
33
34 @online{ctan,
35  title = {Comprehensive TeX Archive Network},
36  shorttitle = {CTAN},
37  date = {1992/},
38  url = {https://www.ctan.org/},
39  urldate = {2021-07-14}
40 }

```

Esimerkki 4.29: Lähdeosteen tietokantatiedosto.

Tietueissa joidenkin kenttien sisältö voi koostua useasta osasta kuten saman teoksen eri tekijöistä. Eri tekijöiden nimet erotetaan author- ja editor-kentissä toisistaan and-sanalla. Oletuksena `biblatex` katsoo, että tekijät ovat henkilöitä, ja käsittelee esimerkiksi etu- ja sukunimet tiettyllä tavalla: jos mukana on pilkku, sitä ennen on sukunimi, ja etunimet tulevat pilkun jälkeen; jos pilkkua ei ole, etunimet ovat ensin, ja sukunimi on lopussa.

Jos kuitenkin teoksen tekijänä on yritys tai yhteisö, täytyy sen nimi kirjoittaa kokonaan aaltosulkeisiin, jottei sitä tulkittaisi henkilön nimeksi. Tällaisten aaltosulkeiden sisällä voi käyttää and-sanaa normaalisti, eikä sitä tulkita eri tekijöiden erottimeksi. Seuraavassa on näistä esimerkit:

```
author = {Meikäläinen, Matti and Teikäläinen, Teija}
author = {{Org. of Latex and Typography} and Meikäläinen, Matti}
```

Muunkinlaisia useasta osasta koostuvia kenttiä on olemassa. Asiasanaketän (`keywords`) eri sanat erotetaan toisistaan pilkulla, ja sivunumeroissa (`pages`) voi olla myös lukualueita, jotka ilmaistaan yhdysmerkillä.

```
keywords = {eri, sanoja, peräkkäin}
pages = {15-19}
```

Teostietokantaan voi määrittää vakiosisältöisiä muuttujia käyttämällä `@string`-rakennetta. Vakioihin voi sitten viitata teostietueiden kentistä esimerkin 4.30 tavoin. Vakiot ovat hyödyllisiä silloin, kun sama kentän sisältö toistuu useissa teoksissa, kuten tässä esimerkissä sama tekijä (`author`) ja aikakauslehden nimi (`journaltitle`). Vakioita voi yhdistää saman kentän muuhun sisältöön käyttämällä #-merkkiä, kuten esimerkin rivillä 13 on tehty.

KÄYTTÖNOTTO

`biblatex`-paketti otetaan käyttöön esimerkin 4.31 rivien avulla. Mukana on myös kielipaketti sekä `csquotes`, joka sisältää lainausmerkkeihin liittyvää logiikkaa (luku 2.1.9). Ilman sitä `biblatex` ei saa eri kielten erilaisia lainausmerkkejä oikein vaan käyttää pelkästään amerikkalaisia (“ ”).

Paketin asetuksissa käytetään valitsinta `style` ja sen asetusta `author-year`, joka asettaa lähdeviittausten ja lähdeluettelon tyyliksi tekijän ja

```

1 @string{
2   itse = {Meikäläinen, Matti},
3   lehti = {Hienon hieno aikakauslehti}
4 }
5
6 @article{hieno_artikkeli,
7   author = itse,
8   journaltitle = lehti,
9   ...
10 }
11
12 @article{toinen_artikkeli,
13   author = itse # { and Teikäläinen, Teija},
14   journaltitle = lehti,
15   ...
16 }

```

Esimerkki 4.30: Muuttujien käyttö ja @string-rakenne.

```

1 % Kielipaketti polyglossia tai babel on ladattava ennen biblalexia.
2 \usepackage{polyglossia} \setdefaultlanguage{finnish}
3 \usepackage[main=finnish]{babel}
4
5 % Kielikohtaiset lainausmerkit oikein csquoten avulla.
6 \usepackage{csquotes}
7
8 \usepackage[style=authoryear]{biblalex}

```

Esimerkki 4.31: biblalex-paketin käyttöönotto ja asetuksia.

vuosiluvun. Se on yleinen käytäntö suomenkielisissä teksteissä. Vastavia tyylejä ovat myös authoryear-comp, authoryear-ibid ja authoryear-icom, jotka lisäksi tiivistävät peräkkäisiä lähdeviittauksia, jos teoksen tekijä on sama.

Numerointiin tai kirjainlyhenteisiin perustuvat lähdeluettelo- ja viittaustyyli ovat nimeltään numeric ja alphabetic. Muitakin tyylejä on olemassa, mutta tämän oppaan esimerkeissä käsitellään tekijä-vuosityyliä.

Paketin omien lähdeluettelo- ja viittaustyylien lisäksi Latex-jakelupaketissa on todennäköisesti mukana myös ulkopuolisten tahojen teke-

miä tyylejä. Tyylikokonaisuus nimeltä `biblatex-ext`³⁸ laajentaa `biblatex`-paketin tavallisten tyylien ominaisuuksia. Laajennustyylien käyttäminen ei vaadi erillisen paketin lataamista, vaan tyylin saa käyttöön yksinkertaisesti vain kirjoittamalla sen nimen `biblatex`-paketin lataamisen yhteydessä. Laajennetut tyyli alkavat kirjaimilla `ext-`, esimerkiksi `ext-authoryear` tai `ext-authoryear-comp`.

Kaikki käyttöön otettavat teostietokantatiedostot täytyy esitellä komennolla `\addbibresource`. Tietokantatiedostoja voi olla useampiakin, ja tietokanta voi olla myös verkko-osoitteen takana oleva tiedosto. `\addbibresource`-komennot täytyy kirjoittaa Latex-lähdedokumentin esittelyosaan.

```
\addbibresource{teokset.bib}
\addbibresource{~/texmf/omat_kirjoitukset.bib}
\addbibresource[location=remote]{http://osoite.netissä/yhteiset.bib}
```

Lähdeluettelo ladotaan dokumenttiin komennolla `\printbibliography`. Komennolle voi antaa valinnaisen argumentin, jonka valitsimilla vaikutetaan esimerkiksi lähdeluettelon otsikon tekstiin tai poistetaan automaattinen otsikointi kokonaan. On myös olemassa erilaisia lähdeteosten rajaamisvalitsimia, joiden avulla voi määrittää, mitä teoksia kyseiseen luetteloon halutaan. Sen avulla voidaan esimerkiksi rajata painetut lähteet yhteen luetteloon, julkaisemattomat toiseen ja verkkolähteet kolmanteen. Seuraavassa on erilaisia esimerkkejä:

```
\printbibliography
\printbibliography[title={Lähteet}]
\printbibliography[heading=none, % Ei automaattista otsikkoa,
type=online] % ja rajataan vain online-tyyppiin.
```

Lähdeluetteloon tulevat mukaan vain ne teokset, joihin on viitattu. Mitään ei siis näy, jos ei ole lähdeviittauksia. Seuraavassa alaluvussa käsitellään lähdeviittauskomentoja ja myös ”näkyvätöntä” viittauskomentoa, jolla teoksia saadaan mukaan luetteloon ilman näkyvää viittausta.

Komento	Esimerkki
<code>\cite</code>	Meikäläinen 2020
<code>\textcite</code>	Meikäläinen (2020)
<code>\parencite</code>	(Meikäläinen 2020)
<code>\citeauthor</code>	Meikäläinen
<code>\citeyear</code>	2020
<code>\citetitle</code>	[teoksen nimi]
<code>\footcite</code>	Meikäläinen 2020 [alaviitteessä]
<code>\nocite</code>	[näkyvätön viittaus]

Taulukko 4.13: `biblatex`-paketin lähdeviittauskomentoja.

LÄHDEVIITTAUKSET

Taulukkoon 4.13 on koottu tavallisimpia `biblatex`-paketin viittauskomentoja. Komennon argumentiksi annetaan teoksen yksilöllinen tunniste. Komennolle voi antaa myös valinnaisen argumentin, jolla kerrotaan täsmentävää tietoa lähdeviittauksesta. Yleensä se on viitattavan teoksen sivunumero. Viittaus näkyy dokumentissa esimerkiksi seuraavalla tavalla:

```
\textcite[27--29]{johdatus} % Viittaus teokseen "johdatus".
```

⇒ Meikäläinen ja Teikäläinen (2020, s. 27–29)

Jos halutaan sisällyttää lähdeluetteloon teoksia, joihin ei ole välttämättä viitattu, käytetään dokumentissa kerran ”näkyvätöntä” viittauskomentoa `\nocite`. Sille annetaan argumentiksi tunnisteet niistä teoksista, jotka halutaan mukaan luetteloon. Argumentti `*` (tähti) valitsee kaikki teokset.

```
\nocite{meikäläinen, teikäläinen} % Nämä teokset mukaan.
\nocite{*} % Kaikki mukaan.
```

LÄHDETIEDOSTOJEN KÄÄNTÄMINEN

Latexin kääntäjäohjelmat Lualatex tai Xelatex eivät yksinään riitä, sillä teostietokanta ei ole tavallinen Latex-muotoinen tiedosto. Tarvitaan myös Latex-jakelun mukana tulevaa komentoa `biber`, joka käsittelee

³⁸ <https://www.ctan.org/pkg/biblatex-ext>

teostietokantaan liittyviä tiedostoja. Lopulta Latex-kääntäjäkin täytyy kutsua kaksi kolme kertaa, jotta kaikki ristiviitteet saadaan kuntoon. Komentojen suoritusjärjestys on seuraavanlainen:

```
lualatex teksti.tex
biber teksti.bcf
lualatex teksti.tex
lualatex teksti.tex
```

Edellisen esimerkin komennoissa voi tiedoston nimistä jättää päätteet pois (.tex, .bcf). lualatex-ohjelman paikalla voi olla myös xelatex. Kääntäminen on vielä helpompaa, kun käyttää latexmk-ohjelmaa (luku 1.3.3), joka osaa automaattisesti suorittaa myös biber-ohjelman ja tarvittavat uudelleen kääntämiset. Yksi komento riittää:

```
latexmk -lualatex teksti.tex % tai: -xelatex
```

LÄHDELUETTELOJEN MITTOJA

Lähdeluettelon ulkoasuun voi vaikuttaa muutaman eri mitan avulla, joista esitellään tässä yhteydessä vain osa. Lähdemerkinnän riippuvan sisennyksen suuruus määräytyy mitan `\bibhang` avulla. Yleensä lienee sopivaa asettaa se samaksi kuin tekstikappaleiden ensimmäisen rivin sisennys `\parindent`.

```
\setlength{\parindent}{1.1em} % tekstikappaleiden 1. rivin sisennys
\setlength{\bibhang}{\parindent}
```

Mitta `\bibitemsep` on lähdemerkintöjen välinen pystysuuntainen tila. Sen avulla voi harventaa lähdeluetteloa, jolloin lähdemerkinnät erottuvat paremmin toisistaan. Mitan `\bibnamesep` avulla voi tehdä suuremman pystysuuntaisen välin lähdemerkintöjen väliin silloin, kun teoksen tekijä vaihtuu (author tai editor). Toisin sanoen tämän mitan avulla voi ryhmitellä saman tekijän teokset tiiviimmin yhteen ja jättää väliä seuraavan tekijän teoksiin. Vastaavanlainen mitta on `\bibinitsep`, jota käytetään silloin, kun lähdemerkinnän aloittava kirjain vaihtuu. Tämän avulla voi ryhmitellä lähdemerkinnät aakkosittain eli tehdä suuremman välin aina lähdemerkinnän alkukirjaimen vaihtuessa.

```
\setlength{\bibitemsep}{.5ex plus .1ex minus .1ex}
\setlength{\bibnamesep}{1em plus .2ex minus .1ex}
\setlength{\bibinitsep}{2em plus .2ex minus .1ex}
```

MUITA ASETUKSIA

Lähdemerkintöjen fonttiin voi vaikuttaa määrittelemällä uudelleen komennon `\bibfont` ja sijoittamalla halutut fontti- tai muut komennot kyseisen komennon määritelmään.

```
\renewcommand{\bibfont}{\sffamily\small}
```

Lähdemerkinnät itsessään muodostetaan automaattisesti tiettyjen tyyliasetusten perusteella. Omiakin tyylejä voi tehdä, mutta yleensä riittää vain yksittäisen asetusten muuttaminen. Niistä käsitellään tässä yhteydessä muutama. Asetusten muuttamiseen tarvitaan yleensä `biblatex`-paketin omia asetuskomentoja.

Lähdeluettelon nimet näkyvät oletusasetuksilla siten, että teoksen ensimmäisen tekijän sukunimi mainitaan ensin (luettelon aakkosjärjestyksen vuoksi) mutta saman teoksen muiden tekijöiden etunimi mainitaan ensin. Tekijöiden nimet näkyvät siis seuraavalla tavalla: ”Meikäläinen, Matti ja Teija Teikäläinen”. Suomessa on kuitenkin tapana kirjoittaa kaikki nimet samalla tavalla ja mainita sukunimi aina ensin. Tämä saadaan toteutettua seuraavilla komennolla:

```
\DeclareNameAlias{default} {family-given}
\DeclareNameAlias{sortname} {family-given}
```

⇒ Meikäläinen, Matti ja Teikäläinen, Teija (2020). – –

Samana teoksen eri tekijöiden nimet erotetaan oletuksena toisistaan pilkuilla paitsi kahden viimeisen nimen välissä on *ja*-sana. Usein on kuitenkin tapana käyttää `&`-merkkiä ainakin lähdeluettelossa. Seuraavat esimerkkikomennot asettavat lähdeluettelon kaikkien nimien erottimeksi `&`-merkin.

```
\DeclareDelimFormat[bib]{multinamedelim} {\space\&\space}
\DeclareDelimFormat[bib]{finalnamedelim} {\space\&\space}
```

⇒ Meikäläinen, Matti & Teikäläinen, Teija & Tutkija, Tuija (2020). – –

Erotinmerkkiasetuksen nimi `multinamedelim` tarkoittaa muiden kuin kahden viimeisen tekijän nimen välissä olevaa erotinta. Kahden viimeisen nimen erotin määritellään asetuksella `finalnamedelim`.

Edellisten esimerkkikomentojen valinnainen argumentti `bib` tarkoittaa, että vaikutetaan vain lähdeluetteloon. Argumentti voi olla myös esimerkiksi `cite`, `textcite` tai `parencite`, jolloin vaikutetaan samanimisillä komennoina tehtyihin lähdeviittauksiin: `\cite`, `\textcite` ja `\parencite`. Katso lähdeviittauskomennot taulukosta 4.13. Seuraava esimerkki vaihtaa `\parencite`-komenttoon asetuksen, niin että kahden viimeisen henkilön nimen väliin ladotaan `&`-merkki. Oletuksena viittauksissa käytetään *ja*-sanaa.

```
\DeclareDelimFormat[parencite]{finalnamedelim}{\space&\space}
```

⇒ (Meikäläinen, Teikäläinen & Tutkija 2020)

Useiden saman teoksen tekijöiden luettelot lyhennetään automaattisesti esimerkiksi muotoon ”Meikäläinen et al.”, ja lyhentämisen säännöt määritellään tiettyjen `max-` ja `min-`alkuisten paketin valitsimien avulla. Lähdeluettelossa teoksen tekijäluetteloon vaikutetaan valitsimilla `max-bibnames` ja `minbibnames`, kun taas lähdeviittausten tekijäluetteloon vaikutetaan valitsimilla `maxcitenames` ja `mincitenames`. Asetukset toimivat siten, että jos enimmäismäärä (`max`) ylittyy, typistetään tekijäluettelo vähimmäismäärään (`min`) ja lisätään ilmaus ”et al.” tms.

Tekijäluettelo ei kuitenkaan välttämättä lyhennetä, jos luettelosta tulisi täsmälleen samanlainen kuin jollakin toisella teoksella. Tähän asiaan puolestaan vaikutetaan valitsimella `unique list`, joka on oletuksena päällä viittaustyyliässä `authoryear`.

```
\usepackage[style=authoryear, maxbibnames=99, minbibnames=3,
maxcitenames=3, mincitenames=1, unique list=true]{biblatex}
```

Kun halutaan näyttää lähdeluettelossa vain tekijän etunimen alkukirjain eikä koko etunimeä, käytetään paketin valitsinta `giveninits`.

```
\usepackage[..., giveninits]{biblatex}
```

⇒ Meikäläinen, M. (2020). --

Lähdeluettelossa näytetään teoksen tekijän nimen kohdalla ajatusviiva, jos tekijä on sama kuin luettelon edelliselläkin teoksella. Mikäli tätä (sinänsä yleistä) käytäntöä ei haluta, täytyy käyttää paketin asetusta `dashed=false`.


```
\usepackage[... , dashed=false]{biblatex}
```

Joskus on tapana latoa lähdeluettelossa tekijöiden nimet esimerkiksi pienversaalilla, jotta ne erottuvat luettelosta paremmin. Tällainen muutos vaatii, että määritellään uudelleen henkilön nimiin liittyvät latomiskomennot `\mkbibnamefamily`, `\mkbibnamegiven`, `\mkbibnameprefix` ja `\mkbibnamesuffix`. Se saadaan automaattiseksi seuraavilla komennoilla:

```
\AtBeginBibliography{%  
  \renewcommand{\mkbibnamefamily}[1]{\textsc{#1}}  
  \renewcommand{\mkbibnamegiven} [1]{\textsc{#1}}  
  \renewcommand{\mkbibnameprefix}[1]{\textsc{#1}}  
  \renewcommand{\mkbibnamesuffix}[1]{\textsc{#1}}  
}
```

⇒ MEIKÄLÄINEN, MATTI & TEIKÄLÄINEN, TEIJA (2020). – –

Lähdeluettelossa yhden lähdemerkinnän eri osien erottimena on piste. Joskus kuitenkin tekijöiden nimien ja vuosiluvun jälkeen halutaan kaksoispiste. Se saadaan toteutettua seuraavalla komennolla:

```
\DeclareDelimFormat[bib]{nametitledelim}{\addcolon\space}
```

⇒ Meikäläinen, Matti (2020): – –

Oletuksena `biblatex` kursivoi book-tyyppisten teosten nimen (`title`). Sen sijaan artikkelikokoelmissa (`incollection`) ja aikakauslehdissä (`article`) kursivoidaan julkaistun kokoelman nimi (`booktitle`) ja aikakauslehden nimi (`journaltitle`). Näissä teostyypeissä viitatus artikkelin nimi (`title`) kirjoitetaan lainausmerkkeihin. Käytäntö tuntuu järkevältä, sillä kursivoituna on aina julkaistu kokonainen teos eikä sen osa. Käytännössähän tiedonlähde joudutaan hakemaan teoksen nimen perusteella. Joku voi silti haluta muuttaa näiden ulkoasua ja esimerkiksi kursivoida aina viittauksen kohteena olevan artikkelin. Seuraavassa on esimerkkikomennot edellä mainittujen lähdeluettelon kenttien muuttamiseen.

```
\DeclareFieldFormat[article,incollection]{title}{\emph{#1}}  
\DeclareFieldFormat[article]{journaltitle}{#1}  
\DeclareFieldFormat[incollection]{booktitle}{#1}
```

Edellä olevissa esimerkkikomennossa on valinnaisena argumenttina ne teostyytit, joihin halutaan vaikuttaa. Jos valinnaisen argumentin jättää pois, vaikutetaan kaikkiin teostyyppisiin, ellei tarkempaa teostyyppikohtaista määritelmää ole olemassa. Ensimmäinen pakollinen argumentti on kentän nimi teostietokannassa, ja toinen pakollinen argumentti on sisältö, joka ladotaan lähdemerkintään kyseisen tiedon kohdalle. Teostietokannasta tulevan kentän sisältö on parametrissa #1.

Mikäli haluaa jonkin teoksen tiedon lainausmerkkeihin tai sulkeisiin, kannattaa käyttää komentoa `\mkbibquote` tai `\mkbibparens`. Ne ymmärtävät ottaa huomioon eri kielten lainausmerkkikäytännöt ja mahdolliset sisäkkäiset sulkeet.

```
\DeclareFieldFormat[incollection]{booktitle}{\mkbibquote{#1}}
```

Oletuksena teoksen vuosiluku tai muu päiväys ladotaan lähdeluetteloon sulkeissa. Joissakin lähdeluettelokäytännöissä sulkeita ei kuitenkaan ole, joten seuraavaksi käsitellään keino sulkeiden poistamiseen. Tavallisessa `biblatex`-paketin lähdeluettelotyylissä `authoryear` ei ole omaa asetusta teoksen päiväyksen ulkoasun muuttamiseen, mutta jos käyttää tyyliä `ext-authoryear` (`tms.`), sekin puute korjaantuu, ja voi käyttää `biblabeledate`-asetusta.

```
\usepackage[style=ext-authoryear]{biblatex}
```

```
\DeclareFieldFormat{biblabeledate}{#1}
```

⇒ Meikäläinen, Matti 2020. – –

Artikkelikokoelmissa (teostyyppi `incollection`) mainitaan oletuksena kokoelman nimi ja toimittajat seuraavassa muodossa: ”Teoksessa: *Hieno artikkelikokoelma*. Toim. Kirjailija, Kaisa”. Ensimmäinen mainitaan julkaisun nimi ja sen jälkeen toimittajien nimet. Suomessa on tapana kirjoittaa nämä tiedot toisinpäin ja laittaa toimittajarooli sulkeisiin. Tällaiset asetukset saa käyttämällä tyyliä `ext-authoryear` (`tms.`), paketin asetusta `innamebeforetitle=true` ja seuraavia komentoja:

```
\usepackage[style=ext-authoryear, innamebeforetitle=true]{biblatex}
```

```
\DeclareFieldFormat{editortype}{\mkbibparens{#1}}
```

```
\DeclareDelimFormat{editortypedelim}{\addspace}
```

⇒ -- Teoksessa: Kirjailija, Kaisa (toim.). *Hieno artikkelikokoelma*. --

Lähdeviittauksissa vuosiluvun ja sivunumeroiden välissä käytetään toisinaan pilkkua ja toisinaan kaksoispistettä. Sivunumeroiden yhteydessä voi olla lyhenne ”s.” tai se voidaan jättää pois. Seuraavilla komennoilla vaikutetaan näihin asetuksiin:

```
\DeclareFieldFormat{postnote}{#1} % Lyhenne ”s.” pois.  
\DeclareDelimFormat{postnotedelim}{\addcolon\space} % Kaksoispiste.
```

```
\textcite[15--16]{tunniste} toteaa artikkelissaan -- --
```

⇒ Meikäläinen (2020: 15–16) toteaa artikkelissaan --

Kun saman teoksen usean tekijän luettelo lyhennetään, käytetään oletuksena latinankielistä ilmausta pois jäävien nimien tilalla: ”Meikäläinen et al.”. Ilmauksen voi muuttaa suomenkieliseksi seuraavalla komennolla:

```
\DefineBibliographyStrings{finnish}{  
  andothers = {ym.},  
}
```

⇒ Meikäläinen ym. (2020)

bibtex³⁹-paketti sisältää valtavan paljon asetuksia ja mahdollisuuksia lähdeluettelon ja -viitteiden ulkoasun säätämiseen. Esimerkiksi komennolla `\DeclareBibliographyDriver` voi ottaa täysin haltuun, miten tietty teostyyppi ladotaan lähdeluettelo. Komennolla `\DeclareSortingTemplate` voi määritellä omia aakkostustapoja. Lisätietoa saa paketin ohjekirjasta.

4.14 Asiahakemistot

Asiahakemistot ovat asiasanojen eli tärkeiden käsitteiden tai henkilöiden luetteloita, ja ne ovat yleensä tietokirjan viimeisillä sivuilla. Kunkin

³⁹ <https://www.ctan.org/pkg/bibtex>

```

1 \documentclass{article}
2 \usepackage{indextools}
3 \makeindex
4
5 \begin{document}
6
7 \index{erilaisia} \index{asiasanoja}
8
9 ...
10
11 \printindex
12
13 \end{document}

```

Esimerkki 4.32: Asiahakemistojen toteutuksen perusteet.

hakemiston asiasanat ovat aakkosjärjestyksessä, ja jokaisen sanan perässä luetellaan sivunumerot, joilla kyseistä asiaa käsitellään. Asiahakemistojen tarkoituksena on helpottaa tiedon löytämistä. Tämän oppaan luku ”Asiahakemistot” alkaa sivulta 257.

Latex tarvitsee asiahakemistojen tekemiseen paketin. Niitä on tehty useampia, mutta parhaita taitavat olla `indextools`⁴⁰ ja `imakeidx`⁴¹, joihin tämä opas keskittyy. Paketit ovat käytännössä lähes samanlaisia: `indextools` perustuu `imakeidx`-pakettiin ja korjaa joitakin sen puutteita.

Asiahakemiston toteutuksen perusajatus on yksinkertainen: Lähdetiedoston alussa komennolla `\makeindex` määritellään asiahakemistot. Dokumentin sisältösiivuilla `\index`-komennolla lisätään haluttuja asiasanoja hakemistoihin. Dokumentin lopussa komennolla `\printindex` ladotaan varsinaiset hakemistot. Näitä komentoja käsitellään tarkemmin erillisissä alaluvuissa, mutta esimerkissä 4.32 ovat perusasiat tiiviisti.

4.14.1 Asiahakemistojen määrittely

Ennen asiasanojen ja -hakemistojen käyttöä täytyy hakemistot määrittellä. Yksinkertaisimmillaan määrittelyksi riittää pelkkä yksi `\makeindex`-komento lähdedokumentin esittelyosassa, kuten esimerkissä 4.32

⁴⁰ <https://www.ctan.org/pkg/indextools>

⁴¹ <https://www.ctan.org/pkg/imakeidx>

Valitsin	Merkitys
name	asiahakemiston tekninen nimi
title	asiahakemistolle ladottava otsikko
intoc	sisällysluetteloon lisääminen
columns	asiahakemiston palstojen lukumäärä
columnsep	asiahakemiston palstojen väli (mitta)
columnseprule	pystyviiva palstojen väliin
program	asiahakemiston tekemisestä vastaava ohjelma: <code>makeindex</code> (oletus), <code>xindy</code> ym.
options	ohjelman komentorivin valitsimia

Taulukko 4.14: `\makeindex`-komennon valitsimia.

on tehty. Jos dokumenttiin tarvitaan useampia asiahakemistoja tai jos haluaa vaikuttaa yksittäisten hakemistojen asetuksiin, täytyy tietää `\makeindex`-komennosta enemmän. Sen sijaan yleisiä, kaikkien hakemistojen latomiseen liittyviä asetuksia käsitellään luvussa 4.14.3.

`\makeindex`-komennolle voi antaa yhden valinnaisen argumentin, joka voi sisältää useita pilkulla toisistaan erotettuja valitsimia ja niiden arvoja. Valitsimia on koottu taulukkoon 4.14.

Valitsin `name` on asiahakemiston tekninen nimi, jota käytetään Latexissa sisäisesti. Valitsinta ei tarvitse käyttää, jos tarvitaan vain yksi asiahakemisto. Tällöin nimeksi tulee sama kuin Latex-lähdedokumentin tiedoston nimi. Jos asiahakemistoja tarvitaan useita, on järkevää nimetä jokainen erikseen `name`-valitsimen avulla. Seuraavassa esimerkissä määritellään kolme eri hakemistoa, jotka voisivat olla Latexin komenoille, mitoille ja laskureille:

```
\makeindex[name=komennot, title=Komennot, columns=2, intoc]
\makeindex[name=mitat, title=Mitat, columns=2, intoc]
\makeindex[name=laskurit, title=Laskurit, columns=2, intoc]
```

`\makeindex`-komennon valitsin `title` nimeää otsikon hakemistolle, joka ladotaan `\printindex`-komennolla (luku 4.14.3). Jos `title`-valitsinta ei ole annettu, otsikko tulee komennosta `\indexname`, joka puolestaan määräytyy kieliasetusten perusteella (luku 3.5). Se on suomen kieliasetuksilla ”Hakemisto”.

Jos valitsin `intoc` on mukana, kyseisen asiahakemiston otsikko lisätään myös sisällysluetteloon. Valitsimella `columns` asetetaan hakemis-

ton palstojen lukumäärää ja valitsimella `columnsep` mitta, joka on palstojen välinen tyhjä tila. Palstojen väliin saa pystyviivan antamalla valitsimen `columnseprule`. Teknisesti palstat toteutetaan `multicol`-paketin avulla, joten sen paketin mitat ovat käytettävissä myös asiahakemistonjen latomisessa. Katso lisätietoa palstoista ja `multicol`-paketista luvusta 4.12.2.

Asiahakemiston järjestelystä vastaa erillinen tietokoneohjelma, joka suoritetaan automaattisesti lähdedokumentin kääntämisen yhteydessä. Ohjelmalle on olemassa pari vaihtoehtoa, ja sen voi valita valitsimella `program`. Tärkeimmät vaihtoehdot ovat `makeindex`⁴² ja `xindy`, joka käytännössä suorittaa `texindy`-nimisen ohjelman. Näillä ohjelmilla on omat ohjekirjansa, joita pääsee lukemaan GNU/Linux-järjestelmissä `man`-komentolla. Jos käyttää vaihtoehtoa `program=xindy`, täytyy Latexin kääntäjälle (`lualatex` tai `xelatex`) antaa komentorivivalitsin `-shell-escape`, joka kytkee päälle erään lisäominaisuuden.⁴³

Valitsimella `options` annetaan edellä mainitulle ohjelmalle komentorivivalitsimet. Tätä tarvitaan lähinnä silloin, kun kirjoittaja on tehnyt oman tyylytiedoston tai muita erikoisempia asetuksia asiahakemistolle. Varsinkin `xindy` on monipuolinen ohjelma, jonka toimintaan voi vaikuttaa monella tavalla. Lisätietoa saa `makeindex`- ja `xindy`-ohjelmien ohjekirjoista.

4.14.2 Asiasanojen lisääminen

Asiasanat lisätään komennolla `\index`, joka sijoitetaan lähdedokumentissa juuri siihen kohtaan, jossa kyseistä asiaa käsitellään. Komento itsessään ei lado mitään; se vain kirjaa asiasanan ja sivunumeron muihin. Komentoa käytetään esimerkiksi seuraavalla tavalla:

```
Yhteen \index{kirjainperhe} kirjainperheeseen kuuluu tavallisesti useita \index{kirjainleikkaus} leikkauksia.
```

`\index`-komentolle voi antaa yhden valinnaisen argumentin. Sitä tarvitaan silloin, kun asiasana halutaan lisätä tiettyyn, määrittelyn yhtey-

⁴² `program=makeindex` on oletusasetus, mutta yleisen oletusasetuksen voi vaihtaa paketin lataamisen yhteydessä. Katso luku 4.14.3.

⁴³ Lisätietoa `tex`-komennon ohjekirjasta, komennolla `man tex`.

dessä `name`-valitsimella nimettyyn hakemistoon (luku 4.14.1). Jos esimerkiksi dokumentissa on määritelty asiahakemisto `\makeindex[name=henkilöt]`, lisättäisiin asiasana kyseiseen hakemistoon seuraavasti:

```
\index[henkilöt]{Waltari, Mika}
```

Tämän luvun muissa esimerkeissä ei yleensä käytetä `\index`-komennon valinnaista argumenttia. Argumentti on kuitenkin tarpeen aina, kun halutaan lisätä asiasanoja tiettyyn, määrittelyn yhteydessä nimettyyn hakemistoon.

Asiahakemistot aakkostetaan automaattisesti, mutta yksittäisten sanojen aakkostustapaan voi vaikuttaa käyttämällä `@`-merkkiä `\index`-komennon argumentissa. Ennen `@`-merkkiä on asiasana siinä muodossa kuin se aakkostetaan; `@`-merkin jälkeen annetaan asiasana siinä muodossa kuin se ladotaan asiahakemistoon.

Seuraavassa esimerkissä asiahakemistoon lisätään kolme eri henkilöä, joiden sukunimet ovat joko Vuori tai Wuori. Kaikki nimet aakkostetaan nimen Vuori mukaisesti, eli *v* ja *w* katsotaan aakkostuksen kannalta samaksi kirjaimeksi. Tämä on tavallinen käytäntö suomalaisten sanojen ja nimien aakkostuksessa.

```
\index{Vuori, Lauri}  
\index{Vuori, Anita@Wuori, Anita}  
\index{Vuori, Yrjö@Wuori, Yrjö}
```

```
⇒ Vuori, Anita, 223  
   Vuori, Lauri, 223  
   Wuori, Yrjö, 223
```

Edellä mainittua aakkostustoimintoa tarvitaan myös silloin, kun hakemiston sanojen latomisessa käytetään Latex-komentoja eli asiasana ei ole pelkkää tekstiä. Seuraavassa esimerkissä asiahakemistoon lisätään Latex-komento `\textbullet`. Se aakkostetaan sanan ”textbullet” mukaan mutta hakemistoon se ladotaan tasalevyisellä fontilla (`\texttt`) ja alkuun lisätään kenoviiva (`\textbackslash`).

```
\index{textbullet@texttt{\textbackslash textbullet}}
```

Käytännössä `\index`-komentoja ei kannata aina kirjoittaa lähdedokumenttiin sellaisenaan, vaan komennon voi sisällyttää jonkin toisen komennon määritelmään. Tästä on esimerkki luvussa 2.2.1.

Asiasanat voi järjestää hakemistoon hierarkkisesti eli aihepiirien mukaisesti. Se toteutetaan `\index`-komennon argumentissa `!`-merkin avulla, jolla erotetaan ylemmätasoiset käsitteet alemmantisoisista. Seuraava esimerkki lisää asiahakemistoon sanan ”fontit” ja sen alle kaksi asiaanaa sivunumeroineen:

```
\index{fontit!kirjainperhe}
\index{fontit!kirjainleikkaus}
```

⇒ fontit
 kirjainleikkaus, 224
 kirjainperhe, 224

Asiasanahierarkiassa voi olla korkeintaan kolme tasoa, ja tasot erotetaan toisistaan `!`-merkillä. Jokainen `!`-merkillä erotettu osa voi sisältää `@`-merkin, jolla voi vaikuttaa sanan aakkostamiseen: ensin mainitaan hakusanasta aakkostamisessa käytetty versio ja `@`-merkin jälkeen annetaan hakemistoon ladottava versio.

Kirjoittaja voi vaikuttaa asiasanojen sivunumeroiden latomiseen käyttämällä `\index`-komennon argumentissa asiasanan perässä `|`-merkkiä ja antamalla sen jälkeen komennon nimen. Seuraavassa esimerkissä asiasanan sivunumero ladotaan kursiivilla:

```
\index{asiasana|textit}
```

⇒ asiasana, 224

`\index`-komennon argumenttiin ei kirjoiteta varsinaista komentoa (`\textit`) vaan ainoastaan komennon nimi `|`-merkin jälkeen. Komento voi olla mikä hyvänsä ja sen pitäisi hyväksyä ainakin yksi argumentti: sivunumero. Asiahakemistoissa on joskus tapana merkitä asiasanan pääasiallinen käsittelysivu eri tavalla, esimerkiksi kursiivilla tai lihavoinnilla, ja tämä `|`-ominaisuus sopii siihen.

Yksi hyödyllinen käyttökohde `|`-ominaisuudelle on viittaus toiseen asiaanaan. On määritelty jo valmiiksi komennot `\see` ja `\seealso`, joita voi käyttää `\index`-komennossa seuraavasti:

```
\index{fontit}
\index{kirjainleikkaus|see{fontit}}
\index{kirjainperhe|seealso{fontit}}
```


⇒ fontit, 224
kirjainleikkaus, *katso* fontit
kirjainperhe, *katso myös* fontit

Molemmat komennot hyväksyvät kaksi argumenttia, mutta edellisessä esimerkissä annettiin vain yksi: fontit. Viimeiseksi argumentiksi tulee automaattisesti aina sivunumero, mutta nämä komennot jättävät sen huomioimatta. Sivunumeroa ei haluta näkyviin, koska tarkoitus on vain lisätä asiahakemistoon viittaus toiseen asiasanaan.

Suomen kieliasetuksilla edellä mainitut komennot latovat kursivoidun ilmauksen *katso* tai *katso myös*. Kielikohtaisen määritelmän voi muuttaa kielipaketteihin polyglossia ja babel kuuluvan \addto-komennon avulla. Seuraavassa esimerkissä vaikutetaan suomen kieliasetuksiin (\captionsfinnish). Komennot täytyy sijoittaa dokumentin esitelyosaan:

```
\addto{\captionsfinnish}{  
  \renewcommand{\seename}{katso}  
  \renewcommand{\alsoname}{katso myös}  
}
```

Toisaalta kirjoittaja voi määritellä ihan oman komentonsa asiasanaviittausten tekemiseen. Oman komennon avulla lopputulokseen voi vaikuttaa enemmän. Esimerkissä 4.33 määritellään komento \katso, jolla voi toteuttaa viittauksia toisiin asiasanoihin. Komento hyväksyy kaksi argumenttia mutta jälkimmäinen eli sivunumero jätetään käsittelemättä.

Tavallisesti \index kirjaa muistiin vain sen sivun, jolla komento esiintyy. Jos komento esiintyy usealla peräkkäisellä sivulla, hakemistoon ladotaan automaattisesti sivualue, esimerkiksi 53–55.

Aina ei kuitenkaan ole mielekästä ilmaista aiheen käsittelyä yksittäisten \index-komentojen avulla. Toisinaan voi olla sopivampaa määrittää aiheen käsittelyn alku- ja loppukohta. Alkukohta ilmaistaan \index-komennon argumentissa merkkien | (avulla ja loppukohta merkkien |) avulla. Jos aloitus ja lopetus sattuvat samalle sivulle, hakemistoon ladotaan vain yksi sivunumero. Muussa tapauksessa ladotaan sivualue. Seuraavassa on esimerkki aiheen käsittelyn alun ja lopun merkitsemiseen:

```

1 \newcommand{\katso}[2]{\textrightarrow\ #1}
2
3 % Seuraavat komennot tarvitaan vain kerran dokumentissa, koska ne vain
4 % viittaavat toisiin asiasanoihin.
5 \index{kirjainleikkaus|katso{fontit}}
6 \index{kirjainperhe|katso{fontit}}
7
8 % Seuraavassa on esimerkki varsinaisista asiasanoista. Komennot
9 % sijoitetaan paikkoihin, joissa aihetta käsitellään.
10 \index{fontit!kirjainleikkaus}
11 \index{fontit!kirjainperhe}

```

⇒ fontit

kirjainleikkaus, 225

kirjainperhe, 225

kirjainleikkaus, → fontit

kirjainperhe, → fontit

Esimerkki 4.33: Oman `\katso`-komennon määrittely. Komennon nimeä käytetään `\index`-komennon argumentissa ja sillä viitataan toisiin asiasanoihin.

```

\index{asiasana|()} % aiheen käsittely alkaa
...
\index{asiasana|)} % aiheen käsittely loppuu

```

⇒ asiasana, 53–55

Edellä mainittujen aloitusmerkkien `|` (perään voi kirjoittaa komennon nimen, jos haluaa vaikuttaa sivunumeroiden latomiseen. Tätä käsiteltiin jo edellä, mutta seuraavassa on käytännön esimerkki:

```

\index{asiasana|(textit)} % aiheen käsittely alkaa
...
\index{asiasana|)} % aiheen käsittely loppuu

```

⇒ asiasana, 53–55

Jos itse asiasanaan täytyy sisällyttää edellä mainittuja `\index`-komennon erikoismerkkejä (`@!`), pitää niiden eteen kirjoittaa yleislainausmerkki (`"`). Se estää seuraavan merkin tulkitsemisen `\index`-komennon erikoismerkkiksi.

Valitsin	Merkitys ja vaihtoehtoja
level	Hakemistojen otsikkokomento: <code>\chapter*</code> , <code>\chapter</code> , <code>\section*</code> , <code>\section</code> ym.
toclevel	Hakemiston taso sisällysluettelossa: <code>chapter</code> , <code>section</code> ym.
noclearpage	Ei automaattista sivunvaihtoa hakemistojen alussa.
firstpagestyle	Hakemiston ensimmäisen sivun sivutyylit: <code>plain</code> , <code>empty</code> ym.

Taulukko 4.15: `\indexsetup`-komennon valitsimia, joilla vaikutetaan kaikkien asiahakemistojen latomiseen.

4.14.3 Hakemiston latominen ja asetukset

Hakemistot ladotaan komennolla `\printindex`. Tyypillisesti komento sijaitsee dokumentin lopussa eli esimerkiksi artikkelin tai tietokirjan viimeisillä sivuilla. Jos komennon suorittaa ilman argumentteja, se lataa oletushakemiston eli sen, jolle ei ole annettu teknistä nimeä `\makeindex`-komennon `name`-valitsimella (luku 4.14.1). Komennolle voi antaa hakasulkeissa yhden valinnaisen argumentin, joka ilmaisee ladottavan hakemiston teknisen nimen, esimerkiksi:

```
\printindex           % Latoo nimeämättömän hakemiston.
\printindex[henkilöt] % Latoo hakemiston nimeltä "henkilöt".
```

Luvussa luku 4.14.1 käsitellään yksittäisten hakemistojen asetuksia, mutta `\indexsetup`-komennolla voi määrittää kaikkia hakemistoja koskevia yleisempiä asetuksia. Komennolle annetaan yksi argumentti, joka voi sisältää useita pilkuilla erotettuja valitsimia ja niiden arvoja. Valitsimia on koottu taulukkoon 4.15, ja seuraavassa on käytännön esimerkki:

```
\indexsetup{level=\section*, noclearpage}
```

Valitsimen `level` arvoksi annetaan tyypillisesti otsikkokomento, joka suoritetaan kunkin hakemiston alussa. Itse otsikon tekstin voi määrittellä `\makeindex`-komennon `title`-valitsimella kullekin hakemistolle erikseen. `level`-valitsimelle oletusarvo on dokumenttiluokakohtainen korkein otsikkotaso eli `\chapter*` tai `\section*`. Jos arvoksi antaa tähdettömän otsikkokomennon kuten `\chapter`, se luo automaattisesti

merkinnän sisällysluetteloon. Silloin ei kannata käyttää `\makeindex`-komennon `intoc`-valitsinta.

Käytännössä `level`-valitsimen arvoksi voi antaa minkä hyvänsä komennon, joka hyväksyy yhden argumentin eli kyseisen hakemiston otsikkotekstin. Kirjoittaja voi siis asettaa tähän oman komentonsa, jonka määritelmässä mahdollisesti suoritetaan muutakin kuin pelkkä otsikkokomento. `toclevel`-valitsimella ilmaistaan hakemistojen taso sisällysluettelossa. Valitsimen arvoksi annetaan otsikkokomentojen nimi kuten `section` tai `subsection`.

Oletuksena jokaisen asiahakemiston alussa suoritetaan sivunvaihtokomento `\clearpage`. Käyttämällä valitsinta `noclearpage` estetään automaattinen sivunvaihto. Valitsimella `firstpagestyle` asetetaan kunkin asiahakemiston ensimmäisen sivun sivutyyli (luku 3.2.4). Oletusarvo on `plain` eli tavallinen, oletuksena pelkän sivunumeron sisältävä sivutyyli. Tämä valitsin ei ole käytössä, jos ladattuna on paketti `fancyhdr`.

Ennen `\printindex`-komentoa voi käyttää komentoa `\indexprologue`, jolla määritetään johdantoteksti asiahakemiston alkuun. Komento vaiuttaa seuraavaksi ladottavaan asiahakemistoon, eli se kannattaa sijoittaa juuri ennen `\printindex`-komentoa. Komentoa käytetään seuraavasti:

```
\indexprologue[väli]{teksti}
```

Pakollinen argumentti `teksti` on hakemiston johdannoksi tarkoitettu teksti. Valinnainen argumentti `väli` on tarkoitettu komennolle, joka latoo pystysuuntaisen välin johdantotekstin jälkeen, ennen asiasanojen luetteloa. Oletuksena väliksi tulee komennon `\bigskip` latoma väli, mutta muunlaisen välin saa kirjoittamalla valinnaiseen argumenttiin esimerkiksi `\vspace`-komennon ja sen argumentiksi halutun mitan.

Asiahakemistopakettien (`indextools` tai `imakeidx`) lataamisen yhteydessä voi määrittää asetuksia tiettyjen valitsimien avulla. Asetusvalitsimet kirjoitetaan `\usepackage`-komennon valinnaiseen argumenttiin ja ne erotetaan toisistaan pilkulla.

Esimerkiksi valitsin `xindy` aiheuttaa sen, että `\makeindex`-komentoilla määritellyille asiahakemistoille tulee oletuksena asetus `program=xindy`. Muutoin oletus on `program=makeindex` (luku 4.14.1).

```
\usepackage[xindy]{indextools}
```

Jos omassa dokumentissa tarvitaan useita asiahakemistoja, saatetaan joskus tarvita valitsinta `splitindex`. Se kiertää erästä Latexin tiedostojen käsittelyn puutetta. Nimittäin Latex ei voi kääntämisen yhteydessä kirjoittaa kovin moneen tiedostoon, ja jokainen asiahakemisto tarvitsisi omansa. Rajoitukset saattavat tulla vastaan useiden asiahakemistojen käsittelyssä.

Käyttämällä valitsinta `splitindex` kierretään edellä mainittua rajoitusta siten, että ensin kaikki asiahakemistot kirjoitetaan yhteen väliaikaistiedostoon ja myöhemmin ne erotetaan automaattisesti toisistaan suorittamalla `splitindex`-niminen ohjelma. Tämän valitsimen käyttö vaatii, että Latexin kääntäjäohjelmalle (`lualatex` tai `xelatex`) annetaan komentorivivalitsin `-shell-escape`.

4.15 Kuvat

Perus Latex ei pysty juuri minkäänlaiseen kuvien eli tietokonegrafiikan käsittelyyn, mutta avuksi on tehty monipuolisia paketteja. Niiden avulla voidaan esimerkiksi ladata ja latoa kuvatiedostoja sekä piirtää vektorigrafiikkakuvia Latex-komentojen avulla.

4.15.1 Kuvatiedostot

Paketti `graphicx`⁴⁴ sisältää toiminnot kuvatiedostojen käsittelyyn sekä tekstin tai muun sisällön kääntelyyn ja skaalaamiseen. Kuvatiedostojen peruskäyttö sujuu komennolla `\includegraphics` seuraavasti:

```
\includegraphics[width=6cm]{kuvatiedosto.jpg}
```

Tällä tavoin lisätty kuvatiedosto käyttäytyy tekstikappaleessa kuin mikä tahansa kirjain tai laatikko. Edellisen esimerkin pakollisessa argumentissa oleva `kuvatiedosto.jpg` on tiedoston nimi. Se voi sisältää myös hakemistopolun, jos tiedosto ei sijaitse samassa hakemistossa Latex-lähdetiedostojen kanssa. Kaikki käyttöjärjestelmän hyväksymät tiedos-

⁴⁴ <https://www.ctan.org/pkg/graphicx>

Valitsin	Merkitys
width	leveysmitta
height	korkeusmitta
angle	kääntökulma asteina
origin	kääntämisen keskipiste: c (keskikohta), l (vasen), r (oikea), t (ylä), b (ala), B (rivin peruslinja)
trim	reunojen siirto: vasen, ala, oikea, ylä
clip	leikkaminen reunojen ulkopuolelta

Taulukko 4.16: `\includegraphics`-komennon valinnaiseen argumenttiin sopivia valitsimia.

tonnimet eivät Latexissa toimi, vaan kannattaa jättää välilyönnit pois ja pitiäytyä suppeassa latinalaisessa merkkivalikoimassa.

`\includegraphics`-komennon valinnaiseen argumenttiin voi kirjoittaa monenlaisia pilkulla toisistaan erotettuja valitsimia ja niiden arvoja. Joitakin valitsimia on koottu taulukkoon 4.16. Tavallisimpia ovat varmaankin `width` ja `height`, joilla määritetään, minkä levyisenä tai korkeusena kuva halutaan latoa. Näistä voi asettaa vain jommankumman, ja toinen mitta määräytyy alkuperäiskuvan mittasuhteiden perusteella. Asettamalla sekä leveyden että korkeuden kuva skaalataan juuri kyseisiin mittoihin.

Kuvan kääntämisessä käytetään `angle`-valitsinta ja astelukua. Kuva kääntyy `origin`-valitsimella määritetyn pisteen kautta, ja valitsimen arvoksi annetaan taulukossa 4.16 mainittuja kirjaimia tai niiden yhdistelmiä. Esimerkiksi `origin=lt` tarkoittaa vasenta (l) ylänurkkaa (t).

Kuvaa voi rajata käyttämällä yhdessä valitsimia `trim` ja `clip`. `trim`-valitsimen arvoksi annetaan neljä välilyönnilä erotettua mitta, joiden järjestys on seuraava: vasen, ala, oikea, ylä. Tämä valitsin vain siirtää kuvan reunoja sisäänpäin, mutta kun mukaan ottaa valitsimen `clip`, kuvasta myös leikataan reunojen ulkopuolinen alue pois. Seuraavassa esimerkissä kuvaa leikataan vasemmalta 1 mm, alhaalta 2 mm, oikealta 3 mm ja ylhäältä 4 mm. Kuvan lopulliseksi leveydeksi tulee 40 mm.

```
\includegraphics[width=40mm,
clip, trim={1mm 2mm 3mm 4mm}]{kuvatiedosto.jpg}
```

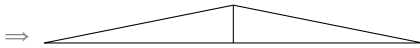
4.15.2 Vektorigrafiikka

Paketti `tikz`⁴⁵ tuo Latexiin monipuolisen apukielen, jolla kirjoittaja voi piirtää vektorigrafiikkaa. Paketin taustalla toimii toinen paketti ja grafiikkajärjestelmä `pgf` eli *portable graphics format*, mutta kirjoittajan ei yleensä tarvitse välittää taustalla olevasta matalan tason tekniikasta.

Tässä oppaassa käsitellään vain joitakin `tikz`-paketin perustoimintoja. Paketin omassa ohjekirjassa on yli 1 300 sivua, eli valtavan paljon hienoja ominaisuuksia jää lukijan itse selvittäväksi. Runsasta sivumäärää ei kannata säikähtää: ei opasta tarvitse kokonaan lukea, ja alkupuolella teksti johdattelee lukijaa eteenpäin helppojen ja käytännöllisten esimerkkien avulla.

Vektorigrafiikka toteutetaan ympäristön `tikzpicture` sisällä. Se ei ole tavallinen Latexin ympäristö, vaan se on tarkoitettu vain piirtokomennoilta, joilla on oma, muusta Latexista hieman poikkeava kielioppinsa. Seuraavassa on pieni esimerkki:

```
\begin{tikzpicture}[x=1mm, y=1mm]
  \draw (0,0) -- (25,5) -- (50,0) -- cycle;
  \draw (25,0) -- (25,5);
\end{tikzpicture}
```



Edellisessä esimerkissä ympäristön alussa määritellään hakasulkeissa perusasetuksia. Tässä asetetaan x- ja y-akselien suuntaiset yksikkömitat 1 mm:n pituiseksi. Piirtokomentojen koordinaatit käyttävät sen jälkeen näitä yksiköitä. Jos yksiköitä ei määritellä, käytetään oletusyksikköä, joka on 10 mm. On myös mahdollista antaa piirtokomentojen koordinaateissa suoraan Latexin mittayksiköt, esimerkiksi (25mm, 5mm) tai (10bp, 20bp), jolloin koordinaatit tulevat juuri näiden mittojen mukaiseksi. Piirustusalueen origo eli koordinaatti (0, 0) sijaitsee vasemmassa alanurkassa, ja koordinaatisto kasvaa ylös ja oikealle.

Edellä olevassa esimerkissä piirretään komennolla `\draw` viivakuvio, jonka välipisteiden koordinaatit ilmaistaan sulkeissa. Lopussa oleva `cycle` piirtää viivan takaisin saman `\draw`-komennon alkupisteeseen.

⁴⁵ <https://www.ctan.org/pkg/tikz>

```

1 \begin{tikzpicture}[x=1mm, y=1mm]
2   \draw (0,0) rectangle (10,5);
3   \draw (20,2.5) circle [radius=2.5];
4   \draw (35,2.5) ellipse [x radius=4, y radius=2, rotate=30];
5   \draw (55,0) arc [start angle=0, end angle=90, radius=5];
6   \draw (65,0) .. controls (75,7) and (80,7) .. (80,0);
7 \end{tikzpicture}

```

⇒



Esimerkki 4.34: Erilaisia geometrisia kuvioita: suorakulmio, ympyrä, ellipsi, ympyrän kaari ja taivutettu viiva.

Tätä kokonaisuutta kutsutaan poluksi, ja sen lopussa täytyy olla puolipiste (;).⁴⁶ Esimerkissä on toinenkin polku eli toinen `\draw`-komento, joka piirtää pystyviivan suuren kolmion keskelle ja jakaa sen kahdeksi pienemmäksi kolmioksi.

Koordinaatit voi ilmaista suhteessa polun edelliseen pisteeseen lisäämällä koordinaattisulkeiden eteen kaksi plusmerkkiä. Seuraavassa esimerkissä havainnollistetaan sitä. Kuvion (polun) neljä ensimmäistä pistettä ilmaistaan x- ja y-koordinaateilla mutta viimeinen piste ilmaistaan kulman (25°) ja pituuden (8) avulla.

```

\begin{tikzpicture}[x=1mm, y=1mm]
\draw (0,0) -- ++(0,5) -- ++(10,0) -- ++(0,-5) -- ++(25:8);
\end{tikzpicture}

```



Viivojen lisäksi on olemassa komentoja muillekin geometrisille peruskuvioille. Esimerkissä 4.34 piirretään suorakulmio, ympyrä, ellipsi, sektorin kaari ($0-90^\circ$) ja taivutettu viiva. Viimeksi mainittu (rivi 6) sisältää alkupisteen, kaksi näkymätöntä ohjauspistettä (`controls`, `and`), joiden suuntaan viivaa taivutetaan, sekä loppupisteen.

Kuten esimerkki 4.34 osoittaa, piirtokomennoille voi antaa hakasulkeissa lisätietoja. Mahdollisuuksia on valtavan paljon: esimerkiksi vä-

⁴⁶ Polkujen luomisen peruskomento on `\path`, joka ei piirrä mitään, ellei sille anna valinnaista argumenttia `draw`. Komento `\draw` on itse asiassa sama kuin `\path[draw]`.


```

1 \begin{tikzpicture}[x=1mm, y=1mm,
2   nuoli/.style={line width=1bp, draw=red}]
3
4   % Nuolia
5   \draw [nuoli, ->] (0,0) -- (10,5);
6   \draw [nuoli, <<- , draw=blue] (10,0) -- (20,5);
7   \draw [nuoli, <->] (20,0) -- (30,5);
8
9   % Täytetty nelikulmio, viivan paksuus ja kulmien pyöristys
10  \draw [draw=black, fill=yellow, line width=1.5bp, rounded
11    corners=3bp] (40,0) -- ++(3,5) -- ++(20,0) -- ++(3,-5) -- cycle;
12 \end{tikzpicture}

```



Esimerkki 4.35: Erilaisia piirtokomentojen argumentteja: nuolenkärjet, värit, viivan paksuus ja kulmien pyöristys.

rit, viivojen paksuus, nuolenkärjet ja kulmien pyöristys ilmaistaan tällaisten lisätietojen avulla. Esimerkki 4.35 havainnollistaa tavallisimpia valinnaisia argumentteja. Väriarvojen `draw` ja `fill` arvoksi voi antaa mitä hyvänsä nimettyjä värejä, joita käsitellään tarkemmin luvussa 4.2.7.

Esimerkissä 4.35 olevan `tikzpicture`-ympäristön valinnaisessa argumentissa määritellään rivillä 2 oma tyyliasetus nimeltä `nuoli`. Asetukseen kuuluu viivan leveys (`line width`) ja väri (`draw`). Näin määriteltyä `nuoli`-tyyliä käytetään rivien 5–7 `\draw`-komentoissa. Tällä tavoin omia tyyliä määrittelemällä säästää todennäköisesti aikaa ja vaivaa, koska tyyliasetuksia voi sen jälkeen muuttaa yhdestä paikasta.

Usein tarvittavat `\draw`-polut ja komentosarjat kannattaneen piilottaa omien kommentojen sisään. Kirjoittaja voi siis tehdä `\newcommand`-komennolla (luku 2.2) ja muilla vastaavilla omia, mahdollisesti kokeamman tason piirtokomentoja, jotka piirtävät usein tarvittavia kuvan osia.

Kuvaan voi sisällyttää myös tekstiä tai muuta Latexin sisältöä. Se tapahtuu helpoimmin komennolla `\node`, joka tekee niin sanottuja solmuja. Komento tarvitsee ainakin solmun sijaintikoordinaatit ja ladottavan sisällön. Esimerkissä 4.36 on kolme erityyppistä solmua.

```

1 \begin{tikzpicture}[x=1mm, y=1mm]
2   \node at (0,0) [draw=red, rectangle, rounded corners=3bp] {vasen};
3   \node [color=blue] at (20,0) {keski};
4   \node at (40,0) [draw, circle, inner sep=0bp, fill=yellow] {oikea};
5 \end{tikzpicture}

```

⇒

vasen

keski

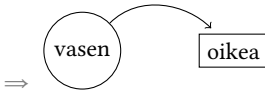


Esimerkki 4.36: Solmuja tehdään `\node`-komennolla.

```

1 \begin{tikzpicture}[x=1mm, y=1mm]
2   \node (ympyrä) at (0,0) [draw, circle] {vasen};
3   \node (suorakulmio) at (20,0) [draw, rectangle] {oikea};
4   \draw [->, shorten >=1mm] (ympyrä) to [out=45, in=135] (suorakulmio);
5 \end{tikzpicture}

```



Esimerkki 4.37: Solmujen nimeäminen ja kytkeminen viivan avulla.

Kuten esimerkistä näkyy, solmulle voi määrittää kehykset ja niiden värin, (`draw`), täyttövärin (`fill`), sisällön värin (`color`), kehyksen kulmien pyöristyksen (`rounded corners`) ja sisäisen tyhjän tilan suuruuden (`inner sep`). Kaikenlaista muutakin voi tehdä valinnaiseen argumenttiin sisällytettävien valitsimien avulla, mutta niistä täytyy lukea lisää paketin omasta ohjekirjasta.

Solmulle voi määrittää yksilöllisen nimen, jota voi sitten käyttää esimerkiksi viivojen piirtämiseen solmujen välille. Ei siis tarvitse käyttää tavallisia `\draw`-komennon koordinaatteja vaan voi käyttää solmulle annettua nimeä. Tätä havainnollistetaan esimerkissä 4.37, jossa esitellään myös `\draw`-komennon `to`-operaatio. Se mahdollistaa solmujen välisen viivan lähtö- ja tulokulman valinnan valitsimilla `out` ja `in`. Esimerkissä myös lyhennetään nuolta loppupäästä käyttämällä valitsinta ”`shorten >`”.

Tämän alaluvun ohjeiden avulla pääsee aika mukavasti alkuun `tikz`-paketin käytössä ja saa toteutettua tavallisimmat vektorigrafikkakuviot. Paketin omaan ohjekirjaan kannattaa silti tutustua, sillä `tikzpicture`-ympäristön mahdollisuudet ovat valtavat.

4.16 Matematiikka

Latexin matematiikkatila on suunniteltu matemaattisten kaavojen latomiseen eli matematiikan syntaksia varten. Se on aivan oma todellisuutensa, joka ei tunnu noudattavan samoja sääntöjä kuin tekstitila. Matematiikkatilassa ovat voimassa eri komennot, eri fontit, erilainen merkien käyttäytyminen ja erilaiset välistykset. Tässä luvussa käsitellään tilan käyttöä ja matemaattisen syntaksin kirjoittamista. Fonttiasetuksia käsitellään luvussa 3.4.12.

4.16.1 Matematiikkatilan käyttö

Matematiikkatila voidaan kytkeä päälle joko tavallisen, tekstitilassa toimivan rivin sisällä tai omassa tekstikappaleessaan. Tekstirivillä matemaattiset kaavat lisätään komentojen `\(` ja `\)` väliin, kahden `$`-merkin väliin tai ympäristön `math` sisälle. Seuraavassa on esimerkki kaikista kolmesta:

Kaava `\(y = 2x + 3 \)` on suoran yhtälö.

Kaava `$y = 2x + 3$` on suoran yhtälö.

Kaava `\begin{math} y = 2x + 3 \end{math}` on suoran yhtälö.

⇒ Kaava `y = 2x + 3` on suoran yhtälö. Kaava `y = 2x + 3` on suoran yhtälö. Kaava `y = 2x + 3` on suoran yhtälö.

Kuten edellä olevasta esimerkistä näkyy, matematiikkatilassa kirjaimet ladotaan kursiivilla. Tavalliset lähdetiedostoon kirjoitetut kirjaimet on tarkoitettu muuttujien nimiksi, kuten tässä esimerkissä x ja y .

Jos matemaattiset kaavat ovat pitkiä tai vievät pystysuuntaista tilaa enemmän kuin tavallisen rivikorkeuden verran, on parasta latoa ne omaksi kappaleekseen. Se toteutetaan esimerkiksi kirjoittamalla kaavat ympäristön `displaymath` sisään tai komentojen `\[` ja `\]` väliin. Molemmat toimivat samalla tavalla.

```
\[ \left(\frac{1}{x}\right)^2 = \frac{1}{x^2} \]
```

$$\left(\frac{1}{x}\right)^2 = \frac{1}{x^2}$$

Ympäristö `equation` toimii muuten samalla tavalla, mutta se latoo kaavan viereen myös järjestysnumeron ristiviittauksia varten. Ympäristön sisälle voi kirjoittaa `\label`-komennon, jonka argumentissa annetaan kaavalle yksilöllinen tunniste. Tekstistä voi viitata kaavaan `\ref`-komennolla, jonka argumenttina on kaavan tunniste. Ristiviittauksia käsitellään tarkemmin luvussa 4.10.

Kaavojen numerot tulevat laskurista `equation`, ja numeron latomiseen voi vaikuttaa määrittelemällä uudelleen komennon `\theequation`. Seuraavassa esimerkissä kaavan numeroon asetetaan ensiksi `\section`-tasaisen otsikon numero, piste erottimeksi ja lopuksi kyseisen kaavan numero.

```
\renewcommand{\theequation}{\thesection.\arabic{equation}}
```

Latexin lähdedokumentissa matematiikkatilassa kaavojen täytyy sisältyä yhteen kappaleeseen eli tyhjiä rivejä ei sallita. Matemaattiset kaavat ladotaan oletuksena sivun keskelle vaakasuunnassa, mutta jos asettaa Latexin dokumenttiluokalle (luku 3.1.2) valitsimen `fleqn`, ne ladotaan sivun vasempaan reunaan. Kaavojen numerot ladotaan oletuksena sivun oikeaan reunaan, mutta vasemmalle ne saa käyttämällä dokumenttiluokan valitsinta `leqno`.

Matematiikkaympäristössä on käytettävissä taulukkoympäristö `array`, joka toimii pitkälti samalla tavalla kuin tekstiilan taulukotkin (luku 4.7). Pelkästään matematiikkatilassa toimiva `array`-ympäristö mahdollistaa useiden kaavarivien latomisen ja kohdistamisen vaakasuunnassa tietylle kohdalle. Esimerkiksi yhtälöt on mielekästä sijoittaa samalle tasalle yhtäsuuruusmerkin kohdalta.

```
\[ \begin{array}{r@{-}l}
4x - 2 & = 6 \\
x & = 2 \\
\end{array} \]
```

$$4x - 2 = 6$$
$$x = 2$$

Pidemmälle kehitettyjä matematiikkaympäristöjä on paketissa `amsmath`⁴⁷. Esimerkiksi `align`- ja `align*`-ympäristöt pystyvät tasaamaan allekkaiset kaavat tietystä kohdasta, eikä taulukon sarakkeita tarvitse erikseen määritellä. Tasauskohta ilmaistaan lähdedokumentissa `&`-merkillä normaalien taulukoiden tavoin. Ympäristön tähtiversio `align*` ei lado kaavan numeroa mukaan. Edellä olleen yhtälöesimerkin voi toteuttaa yksinkertaisemmin seuraavasti:

```
\begin{align*}
  4x - 2 &= 6 \\
  x      &= 2
\end{align*}
```

Edellä mainittuja `align`- ja `align*`-ympäristöjä ei kirjoiteta komentojen `\[` ja `\]` sisään, eli nämä ympäristöt on tarkoitettu teksti-tilassa käytettäväksi. Ympäristön sisältö on matematiikkatilassa.

Paketti `amsmath` sisältää paljon muitakin hyödyllisiä ympäristöjä ja komentoja matematiikan latomiseen. Paketin ohjekirjaan on erittäin suositeltavaa tutustua.

4.16.2 Matematiikkatilan kielioppia

Matematiikan kirjoittaminen Latexissa on tehty varsin luonnolliseksi. Esimerkiksi tavalliset operaattorit ja kokonaisluvut kirjoitetaan näppäimistöltä sellaisenaan, ja jos desimaalierottimena on piste, sekin syötetään näppäimistöltä suoraan. Suomessa käytetään desimaalierottimena kuitenkin pilkkua, joka toimii Latexin matematiikkatilassa välimerkinä: sen jälkeen ladotaan pieni väli. Suomalaisen desimaalierottimen saa kirjoittamalla pilkun ympärille aaltosulkeet.

```
\[ \pi \approx 3{,}142 \]
```

$$\pi \approx 3,142$$

Toinen vaihtoehto pilkun käyttämiseen desimaalierottimana on ladata paketti `icomma`⁴⁸. Paketti määrittelee matematiikkatilan pilkun toimimaan jokseenkin älykkäästi: jos lähdetiedostossa on pilkun jälkeen

⁴⁷ <https://www.ctan.org/pkg/amsmath>

⁴⁸ <https://www.ctan.org/pkg/icomma>

väli, pilkku ladotaan välimerkkinä; jos väliä ei ole, pilkku katsotaan desimaalierottimeksi.

Plus-, jako- ja yhtäsuuruusmerkki sekä pienempi kuin ja suurempi kuin -merkit (+ : / = < >) kirjoitetaan näppäimistöltä sellaisenaan. Miinusmerkki (−) kirjoitetaan yhdysmerkin (-) avulla, eli yhdysmerkki ladotaan dokumenttiin Unicode-merkistön miinusmerkkinä U+2212 MINUS SIGN. Kertomerkit voi kirjoittaa lähdedokumenttiin sellaisenaan mutta myös komennoilla `\cdot` (·) ja `\times` (×).

Pienet sulkeetkin voi kirjoittaa näppäimistöltä suoraan, mutta matematiikassa tarvitaan usein erikokoisia, tilanteeseen mukautuvia sulkeita. Ne tehdään komentojen `\left` (vasen) ja `\right` (oikea) avulla. Komentojen jälkeen kirjoitetaan haluttu suljemerkki, esimerkiksi `\left` (ja `\right`). Itseisarvoa merkitsevät pystyviivat tehdään samoilla komennoilla, mutta suljemerkkien tilalle kirjoitetaan pystyviiva (|).

```
\[ \left| a + \left( \frac{b}{c(d-1)} \right) \right| \right| \]
```

$$\left| a + \left(\frac{b}{c(d-1)} \right) \right|$$

Komentoja `\left` ja `\right` täytyy käyttää pareittain, jotta Latex osaa laatoa oikeankokoiset sulkeet. Jos ei halua suljemerkillä paria, kirjoitetaan toisen suljekomennon ”sulkeeksi” piste. Seuraavassa esimerkissä ladotaan vasemmalle aaltosulje (`\left\{`) mutta oikealle ei mitään (`\right.`).

```
\[ \left\{ \begin{array}{r@{~}l}
x &= 4 \\
y &= -2
\end{array} \right. \]
```

$$\begin{cases} x = 4 \\ y = -2 \end{cases}$$

Matematiikatilan ylä- ja alaindeksit toteutetaan sirkumfleksin (^) ja alaviivan (.) avulla. Välittömästi merkin jälkeen oleva merkki ladotaan ylä- tai alaindeksiksi, ja jos indeksi sisältää enemmän kuin yhden merkin, kirjoitetaan kokonaisuus aaltosulkeisiin. Joidenkin matemaattisten operaattorikomentojen yhteydessä indeksit ladotaan poikkeuksellisella tavalla, esimerkiksi kokonaan operaattorin ylä- tai alapuolelle.

`\[x^{n-1} \quad x_i \quad \sum_{i=1}^n \quad \int_0^\infty \quad \lim_{n \rightarrow \infty} \quad \int_{-\infty}^{\infty} \quad \lim_{n \rightarrow \infty}`

$$x^{n-1} \quad x_i \quad \sum_{i=1}^n \quad \int_0^\infty \quad \lim_{n \rightarrow \infty}$$

Lausekkeen osia voi ryhmitellä ylä- tai alapuolisella aaltosulkeella, jotka tehdään komennoilla `\overbrace` ja `\underbrace`. Jos näiden komentojen jälkeen käyttää ylä- tai alaindeksiä, se ladotaan sulkeen keskelle. Seuraavassa on esimerkki:

$$\left[\overbrace{3x + x}^{4x} - \underbrace{5y - 2y}_{3y} \right] = 4x - 3y$$

Muuttujien, funktion nimien tai muiden symbolien jäljessä oleva ' -merkki tehdään yleisheittomerkin (') avulla. Matematiikassa se voi tarkoittaa esimerkiksi funktion derivaattafunktiota, ja Unicode-merkkistössä merkin tunnus `U+2032 PRIME`.

$$\left[f(x) = 3x^2 - 2x \quad f'(x) = 6x - 2 \right]$$

$$f(x) = 3x^2 - 2x \quad f'(x) = 6x - 2$$

Murtoluvuille ja jakoviivan latomiseen on komento `\frac`, jolle annetaan kaksi argumenttia: osoittaja ja nimittäjä. Neliö- ja muut juuret tehdään `\sqrt`-kennolla, jolle annetaan ainakin yksi argumentti. Kennolle voi antaa hakasulkeissa toisenkin argumentin, joka ilmaisee juuriluvun. Seuraavassa on esimerkki murtoluvun, neliöjuuren ja kuu-tiojuuren toteuttamisesta:

$$\left[\frac{1}{3x + 1} \quad \sqrt{9} = \sqrt[3]{27} \right]$$

Vektorien nimien latomiseen eli merkkien yläpuoliselle nuolelle on oma komento `\vec`, joka toimii yhden kirjaimen kanssa. Kahden merkin mit-taiseen nuoleen tarvitaan komentoa `\overrightarrow` (tai `\overleftarrow` `\overrightarrow`).

Komento	Esimerkki	Merkitys
<code>\mathrm</code>	ABC abc	antiikva, serif, roman
<code>\mathsf</code>	ABC abc	groteski, sans serif, gothic
<code>\mathtt</code>	ABC abc	tasalevyinen, typewriter
<code>\mathcal</code>	<i>ABC</i>	kalligrafinen
<code>\mathbf</code>	ABC abc	lihavoitu, bold
<code>\mathit</code>	<i>ABC abc</i>	kursiivi, italic

Taulukko 4.17: Tekstin latominen matematiikkatilassa vaatii erityisen komennon.

```
[ \vec{a} \quad \overrightarrow{AB} ]
```

$$\vec{a} \quad \overrightarrow{AB}$$

Matriisit voi toteuttaa luvussa 4.16.1 esitellyn `array`-ympäristön avulla, mutta kätevämpää on käyttää `amsmath`-paketin ympäristöjä. Kullekin erilaiselle suljetyypille on oma ympäristönsä: `matrix`, `pmatrix` (`()`), `bmatrix` (`[]`), `Bmatrix` (`{}`), `vmatrix` (`|`) ja `Vmatrix` (`||`). Ympäristön sisällä matriisin rivin solut erotetaan toisistaan samoin kuin taulukoissakin eli `&`-merkillä ja rivinvaihto tehdään `\\`-komennolla. Seuraavassa on esimerkki kahdesta eri ympäristöstä:

```
[ \begin{matrix} 1 & 2 \\ 3 & 4 \end{matrix} \quad \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix]
\begin{matrix}
1 & 2 & 3 \\
4 & 5 & 6 \\
7 & 8 & 9
\end{matrix}
```

$$\begin{matrix} 1 & 2 \\ 3 & 4 \end{matrix} \quad \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

Kuten on jo todettu, tavalliset kirjaimet on matematiikkatilassa tarkoitettu muuttujien nimiksi. Kun täytyy latoa varsinaista tekstiä – teksti-tilan tavoin – täytyy käyttää erityisiä komentoja. Taulukkoon 4.17 on koottu matematiikkatilan tekstinlatomiskomentoja. Komentojen argumenttina oleva teksti ladotaan kuin teksti-tilassa.

Vaakasuuntaisten välien tekemiseen on matematiikkatilassa muutama komento. Komento `\quad` latoo typografisen neliön (1 em) levyisen välin. Se on sama kuin nykyisen fontin koko. Komento `\qquad` latoo

α	<code>\alpha</code>	A	<code>\Alpha</code>	β	<code>\beta</code>	B	<code>\Beta</code>
γ	<code>\gamma</code>	Γ	<code>\Gamma</code>	δ	<code>\delta</code>	Δ	<code>\Delta</code>
ϵ	<code>\epsilon</code>	ε	<code>\varepsilon</code>	E	<code>\Epsilon</code>	ζ	<code>\zeta</code>
Z	<code>\Zeta</code>	η	<code>\eta</code>	H	<code>\Eta</code>	θ	<code>\theta</code>
ϑ	<code>\vartheta</code>	Θ	<code>\Theta</code>	ι	<code>\iota</code>	I	<code>\Iota</code>
κ	<code>\kappa</code>	K	<code>\Kappa</code>	λ	<code>\lambda</code>	Λ	<code>\Lambda</code>
μ	<code>\mu</code>	M	<code>\Mu</code>	ν	<code>\nu</code>	N	<code>\Nu</code>
ξ	<code>\xi</code>	Ξ	<code>\Xi</code>	π	<code>\pi</code>	ϖ	<code>\varpi</code>
Π	<code>\Pi</code>	ρ	<code>\rho</code>	ϱ	<code>\varrho</code>	P	<code>\Rho</code>
σ	<code>\sigma</code>	ς	<code>\varsigma</code>	Σ	<code>\Sigma</code>	τ	<code>\tau</code>
T	<code>\Tau</code>	υ	<code>\upsilon</code>	Y	<code>\Upsilon</code>	ϕ	<code>\phi</code>
φ	<code>\varphi</code>	Φ	<code>\Phi</code>	χ	<code>\chi</code>	X	<code>\Chi</code>
ψ	<code>\psi</code>	Ψ	<code>\Psi</code>	ω	<code>\omega</code>	Ω	<code>\Omega</code>

Taulukko 4.18: Kreikkalaisten kirjainten latominen matematiikkatilassa.

\grave{a}	<code>\grave{a}</code>	\ddot{a}	<code>\ddot{a}</code>	\hat{a}	<code>\hat{a}</code>
\acute{a}	<code>\acute{a}</code>	\check{a}	<code>\check{a}</code>	\widehat{aaa}	<code>\widehat{aaa}</code>
\breve{a}	<code>\breve{a}</code>	\dot{a}	<code>\dot{a}</code>	\tilde{a}	<code>\tilde{a}</code>
\bar{a}	<code>\bar{a}</code>	\mathring{a}	<code>\mathring{a}</code>	\widetilde{aaa}	<code>\widetilde{aaa}</code>

Taulukko 4.19: Matematiikkatilan tarkekomentoja.

2 em:n levyisen välin. Pienempiä välejä saa komennoilla `\`, ($\frac{3}{18}$ em), `\:` ($\frac{1}{18}$ em) ja `\;` ($\frac{5}{18}$ em). Komento `\!` puolestaan tuottaa negatiivisen välin, jonka mitta on $-\frac{3}{18}$ em. Negatiivista väliä voi käyttää liian suuren välin pienentämiseen.

4.16.3 Erikoismerkkejä

Matematiikkatilassa voi käyttää Unicode-merkistöä, eli monet matemaattiset symbolit voi kirjoittaa lähdedokumenttiin sellaisenaan. Voi silti olla helpompaa käyttää erityisiä komentoja sellaisten kirjainten ja symbolien kirjoittamiseen, joita ei ihan helposti pysty tuottamaan näppäimistöltä tai tekstieditorin toimintojen avulla. Matematiikkatilan erikoismerkkejä on koottu oheisiin taulukoihin (4.18–). Lisää symboleja on packageissa `latexsym`⁴⁹ ja `amsmath`⁵⁰.

⁴⁹ <https://www.ctan.org/pkg/latexsym>

⁵⁰ <https://www.ctan.org/pkg/amsmath>

\overrightarrow{AB}	<code>\overrightarrow{AB}</code>	\overleftarrow{AB}	<code>\overleftarrow{AB}</code>
$\underline{\overrightarrow{AB}}$	<code>\underrightarrow{AB}</code>	$\underline{\overleftarrow{AB}}$	<code>\underleftarrow{AB}</code>
\overleftrightarrow{AB}	<code>\overleftrightarrow{AB}</code>	\overleftrightarrow{AB}	<code>\overleftrightarrow{AB}</code>
\overline{AB}	<code>\overline{AB}</code>	\underline{AB}	<code>\underline{AB}</code>
\overbrace{abc}	<code>\overbrace{abc}</code>	\underbrace{abc}	<code>\underbrace{abc}</code>

Taulukko 4.20: Ylä- ja alamerkintöjä.

$<$	<code><</code>	$>$	<code>></code>	$=$	<code>=</code>	$:$	<code>:</code>
\leq	<code>\leq, \le</code>	\geq	<code>\geq, \ge</code>	\ll	<code>\ll</code>	\gg	<code>\gg</code>
\equiv	<code>\equiv</code>	\doteq	<code>\doteq</code>	\prec	<code>\prec</code>	\preceq	<code>\preceq</code>
\succ	<code>\succ</code>	\succeq	<code>\succeq</code>	\sim	<code>\sim</code>	\simeq	<code>\simeq</code>
\subset	<code>\subset</code>	\subseteq	<code>\subseteq</code>	\supset	<code>\supset</code>	\supseteq	<code>\supseteq</code>
\approx	<code>\approx</code>	\cong	<code>\cong</code>	\sqsubset	<code>\sqsubset</code>	\sqsubseteq	<code>\sqsubseteq</code>
\sqsupset	<code>\sqsupset</code>	\sqsupseteq	<code>\sqsupseteq</code>	\bowtie	<code>\bowtie</code>	\in	<code>\in</code>
\ni	<code>\ni</code>	\propto	<code>\propto</code>	\vdash	<code>\vdash</code>	\dashv	<code>\dashv</code>
\models	<code>\models</code>	\mid	<code>\mid</code>	\parallel	<code>\parallel</code>	\perp	<code>\perp</code>
$($	<code>\smile</code>	$)$	<code>\frown</code>	\asymp	<code>\asymp</code>	\notin	<code>\notin</code>
\neq	<code>\neq, \ne</code>						

Taulukko 4.21: Relaatioita. Negaation saa kirjoittamalla komennon eteen `\not`.

$+$	<code>+</code>	$-$	<code>-</code>	\pm	<code>\pm</code>	\mp	<code>\mp</code>
\cdot	<code>\cdot</code>	\div	<code>\div</code>	\times	<code>\times</code>	\setminus	<code>\setminus</code>
\cup	<code>\cup</code>	\cap	<code>\cap</code>	\sqcup	<code>\sqcup</code>	\sqcap	<code>\sqcap</code>
\vee	<code>\vee</code>	\wedge	<code>\wedge</code>	\oplus	<code>\oplus</code>	\ominus	<code>\ominus</code>
\odot	<code>\odot</code>	\oslash	<code>\oslash</code>	\uplus	<code>\uplus</code>	\otimes	<code>\otimes</code>
\sum	<code>\sum</code>	\prod	<code>\prod</code>	\coprod	<code>\coprod</code>	\int	<code>\int</code>
\oint	<code>\oint</code>	\lim	<code>\lim</code>	\bigoplus	<code>\bigoplus</code>	\bigotimes	<code>\bigotimes</code>
\bigodot	<code>\bigodot</code>	\bigvee	<code>\bigvee</code>	\bigwedge	<code>\bigwedge</code>	\bigcup	<code>\bigcup</code>
\bigcap	<code>\bigcap</code>	\biguplus	<code>\biguplus</code>	\bigsqcup	<code>\bigsqcup</code>		

Taulukko 4.22: Operaattoreita.

\leftarrow	<code>\leftarrow, \gets</code>	\rightarrow	<code>\rightarrow, \to</code>
\longleftarrow	<code>\longleftarrow</code>	\longrightarrow	<code>\longrightarrow</code>
\leftrightarrow	<code>\leftrightarrow</code>	\longleftrightarrow	<code>\longleftrightarrow</code>
\Lleftarrow	<code>\Lleftarrow</code>	\Rrightarrow	<code>\Rrightarrow</code>
\Llongleftarrow	<code>\Llongleftarrow</code>	\Rlongrightarrow	<code>\Rlongrightarrow</code>
\Lleftrightarrow	<code>\Lleftrightarrow</code>	\Llongleftrightarrow	<code>\Llongleftrightarrow</code>
\mapsto	<code>\mapsto</code>	\longmapsto	<code>\longmapsto</code>
\hookrightarrow	<code>\hookrightarrow</code>	\hookleftarrow	<code>\hookleftarrow</code>
\lharpoonup	<code>\lharpoonup</code>	\rharpoonup	<code>\rharpoonup</code>
\lharpoonright	<code>\lharpoonright</code>	\rharpoonright	<code>\rharpoonright</code>
\rightharpoonup	<code>\rightharpoonup</code>	\leftharpoonup	<code>\leftharpoonup</code>
\rightleftharpoons	<code>\rightleftharpoons</code>	\iff	<code>\iff</code>
\Uparrow	<code>\Uparrow</code>	\Downarrow	<code>\Downarrow</code>
\updownarrow	<code>\updownarrow</code>	\Updownarrow	<code>\Updownarrow</code>
\nearrow	<code>\nearrow</code>	\searrow	<code>\searrow</code>
\swarrow	<code>\swarrow</code>	\nwarrow	<code>\nwarrow</code>

Taulukko 4.23: Nuolia.

\langle	<code>\langle</code>	\rangle	<code>\rangle</code>	\lfloor	<code>\lfloor</code>	\rfloor	<code>\rfloor</code>
\lceil	<code>\lceil</code>	\rceil	<code>\rceil</code>	\lvert	<code>\lvert</code>	\rvert	<code>\rvert</code>
$\left($	<code>\left(</code>	$\right)$	<code>\right)</code>	$\} $	<code>\}</code>	$\} $	<code>\}</code>
$/$	<code>/</code>	\backslash	<code>\backslash</code>				

Taulukko 4.24: Sulkeita ja erotinmerkkejä.

\neg	<code>\neg, \lnot</code>	\angle	<code>\angle</code>	\emptyset	<code>\emptyset</code>
∞	<code>\infty</code>	\prime	<code>\prime</code>	\prime	<code>\prime</code>
\forall	<code>\forall</code>	\exists	<code>\exists</code>	\wr	<code>\wr</code>
\perp	<code>\perp</code>	\top	<code>\top</code>	\surd	<code>\surd</code>
\dots	<code>\dots</code>	\cdots	<code>\cdots</code>	\vdots	<code>\vdots</code>
\ddots	<code>\ddots</code>	\triangle	<code>\triangle</code>	\triangleleft	<code>\triangleleft</code>
\triangleright	<code>\triangleright</code>	∇	<code>\nabla</code>	\star	<code>\star</code>
\ast	<code>\ast</code>	\circ	<code>\circ</code>	\bigcirc	<code>\bigcirc</code>
\bullet	<code>\bullet</code>	\diamond	<code>\diamond</code>	\amalg	<code>\amalg</code>
\triangle	<code>\bigtriangleup</code>	∇	<code>\bigtriangledown</code>	\dagger	<code>\dagger</code>
\ddagger	<code>\ddagger</code>	\diamond	<code>\diamondsuit</code>	\heartsuit	<code>\heartsuit</code>
\clubsuit	<code>\clubsuit</code>	\spadesuit	<code>\spadesuit</code>	\flat	<code>\flat</code>
\natural	<code>\natural</code>	\sharp	<code>\sharp</code>	\hbar	<code>\hbar</code>
\imath	<code>\imath</code>	\jmath	<code>\jmath</code>	ℓ	<code>\ell</code>
\Re	<code>\Re</code>	\Im	<code>\Im</code>	\aleph	<code>\aleph</code>
\wp	<code>\wp</code>				

Taulukko 4.25: Sekalaisia symboleja.

Luku 5

Erikoisdokumentit

Kaikki dokumentit ja painotuotteet eivät ole samanlaista tekstivirtaa, joka koostuu jäsenelystä rakenteesta, väliotsikoista ynnä muusta selaisesta. Tässä luvussa käsitellään dokumentteja, jotka vaativat osittain toisenlaista rakennetta ja tekniikkaa kuin luvussa 4 on käsitelty.

5.1 Esitysgrafiikka

Esitelmien ja muiden puhe-esitysten tukena käytetään usein esitysgrafiikkaohjelmia kuten Microsoft Powerpointia tai Libreoffice Impressiä. Myös Latex sopii esitysgrafiikan tekemiseen, ja se voi olla osaavissa käsissä jopa kaikkein nopein ja typografisesti tyylikkään työkalu esitysdioiden tuottamiseen.

Latexin tunnetuin ja monipuolisin esitysgrafiikkajärjestelmä on `beamer`¹, joka on oma dokumenttiluokkansa. Sen ohjekirjassa on reilu kaksisataa sivua, joten tässä oppaassa voidaan esitellä vain perusasioita, joilla pääsee alkuun.

5.1.1 Diaesityksen rakentaminen

Esimerkissä 5.1 on kokonainen Latexin lähdetiedosto ja diaesityksen runko. Dokumenttiluokan `beamer` lataamisen yhteydessä (rivi 1) määritellään valitsimella `aspectratio`, mitkä ovat dioiden mittasuhteet. Tässä

¹ <https://www.ctan.org/pkg/beamer>

```

1 \documentclass[aspectratio=169,t]{beamer}
2 \usepackage{polyglossia} \setdefaultlanguage{finnish}
3 \usepackage{hyperref}
4
5 \title{Esitysgraafiikka Latexilla}
6 \author{Teppo Tekijä}
7 \institute{Ladontatieteiden tiedekunta}
8 \date{13.9.2021}
9
10 \begin{document}
11
12 \section{Otsikkodia}
13
14 \begin{frame}
15   \maketitle
16 \end{frame}
17
18 \section{Ensimmäinen väliotsikko}
19
20 \begin{frame}{Pääotsikko}{Alaotsikko}
21
22   \pause Seuraavassa luetellaan jotakin: \pause
23
24   \begin{enumerate}
25     \item ensimmäinen \pause
26     \item toinen \pause
27     \item kolmas
28   \end{enumerate}
29
30 \end{frame}
31
32 \section{Toinen väliotsikko}
33
34 \begin{frame}[c]
35
36   Sisältö on keskitetty pystysuunnassa.
37
38 \end{frame}
39
40 \end{document}

```

Esimerkki 5.1: beamer-dokumenttiluokan avulla tehdyn diaesityksen runko.

esimerkissä arvoksi annetaan 169, joka tarkoittaa leveyden ja korkeuden suhdetta 16:9. Oletusarvo on 43 eli kuvasuhde 4:3. Muita mahdollisia arvoja ovat esimerkiksi 1610 (16:10) ja 54 (5:4).

Esimerkin ensimmäisellä rivillä oleva toinen valitsin t (top) tarkoittaa, että diojen sisältö sijoitetaan dian yläreunaan. Oletusasetus on c (center), joka keskittää dian sisällön pystysuunnassa. Tämän asetuksen voi muuttaa kuhunkin diaan käyttämällä `frame`-ympäristön valinnaista argumenttia, kuten esimerkin rivillä 34 on tehty.

Riveillä 5–8 määritellään dokumentin perustiedot (luku 4.4.1), mutta perus Latexiin verrattuna mukana on uusi komento `\institute`, jolla voi määritellä tekijän edustaman laitoksen. Nämä perustiedot ladotaan näkyviin vasta `\maketitle`-komennolla, joka on esimerkin rivillä 15.

Jokainen yksittäinen sivu eli dia täytyy kirjoittaa `frame`-ympäristön sisään. Ympäristö siis aloittaa aina puhtaan sivun. Ympäristön alussa voi olla aaltosulkeissa sivun pääotsikko ja toisissa aaltosulkeissa alaotsikko. Esimerkin rivillä 20 on tehty juuri näin. Toisen tai molemmat otsikot voi jättää poisikin.

Sivun eli `frame`-ympäristön sisällä voi olla tavallista tekstiä ja käyttää suunnilleen normaaleja Latexin komentoja. Esimerkissä 5.1 on käytetty muutaman kerran `\pause`-komentoa, joka tekee esitykseen tauon. Käytännössä se jakaa sivun sisällön erillisiksi pdf-sivuiksi, minkä avulla sisällön voi paljastaa yleisölle vaihe vaiheelta. Esimerkin riveillä 24–28 käytetään luetelmaa eli `enumerate`-ympäristöä. Se toimii suunnilleen samoin kuin perus Latexin vastaava ympäristö (luku 4.6.1), mutta `beamer`-versiossa on hieman enemmän ominaisuuksia.

Tästä esimerkistä puuttuu hyödyllinen `block`-ympäristö, jolla saa dian sisälle pienen väliotsikon ja siihen kuuluvan tekstikokonaisuuden. Otsikon teksti annetaan ympäristön argumentissa:

```
\begin{block}{Väliotsikko}
...
\end{block}
```

Otsikkokomennot `\chapter`, `\section`, `\subsection` ym. täytyy kirjoittaa `frame`-ympäristöjen ulkopuolelle. `beamer`-luokassa otsikkokomennot eivät lado mitään itse dokumenttiin, vaan ne ainoastaan muodostavat pdf-tiedoston sisällysluettelon. Tosin sisällysluettelon kaltaista ra-

kennetietoa on mahdollista saada näkyviin itse diojen reunoillekin. Niiden tarkoituksena on helpottaa esimerkiksi pitkän diaesityksen seuraamista ja jäsentämistä. Näistä ominaisuuksista voi lukea lisää `beamer`-luokan ohjekirjasta.

Dioihin voi lisätä kuvatiedostoja tai vektorigrafikkaa normaalin `LaTeXin` tavoin eli luvun 4.15 ohjeilla. Palstoja ei kuitenkaan kannata toteuttaa luvun 4.12 keinoilla vaan `beamer`-luokan oman `columns`-ympäristön avulla.

5.1.2 Ulkoasuteemat

`beamer`-dokumenttiluokka sisältää valmiita ulkoasuteemoja eli ulkoasuun vaikuttavia asetusten kokonaisuuksia. Käyttämällä valmiista teemaa saa helposti käyttöönsä jonkun henkilön suunnitteleman tyylikkään kokonaisuuden.

Yleinen teemojen valintakomento on `\usetheme`. Sillä valitaan teema, joka voi määritellä vähän kaikkea diaesityksen ulkoasuun liittyvää: fontit, värit, sisällön asettelua ja diojen reunoille ladottavaa lisätietoa. Alemmantasoisilla teemakomennoilla valitaan vain jonkin pienemmän osa-alueen teema. Esimerkiksi komennolla `\usecolortheme` ja `\usefonttheme` valitaan vain väri- tai fonttiteema.

Fontit tai kirjainperheet sinänsä määritellään ja otetaan käyttöön samalla tavalla kuin `Latexissa` muutenkin (luku 3.4), mutta `beamer`-luokan teema voi määritellä, mitä kirjainperhettä tai -leikkausta käytetään missäkin tilanteessa, esimerkiksi diojen otsikoissa tai leipätekstissä. Näiden asetusten muuttaminen voi vaatia `beamer`-luokan omia komentoja, joista annetaan tietoa luvussa 5.1.3.

Alemmantasoisia teemakomentoja ovat myös `\useinnertheme` ja `\useoutertheme`. Niillä vaikutetaan erilaisiin diojen sisältöelementtien (inner) ulkoasuun ja diojen reunoille (outer) ladottavaan lisätietoon.

Taulukkoon 5.1 on koottu `beamer`-luokan teemakomennot ja valmiita teemoja. Taulukon ensimmäisessä sarakkeessa ovat teeman valintakomennot ja toisessa sarakkeessa teemojen nimiä, joita voi antaa komennot argumentiksi. Ensimmäisenä mainittu `default`-teema on käytössä oletuksena. Teemat otetaan käyttöön kirjoittamalla lähdedokumentin esittelyosaan teemakomentoja, esimerkiksi seuraavalla tavalla:

Komento	Valmiita teemoja
<code>\usetheme</code>	default, boxes, Bergen, Boadilla, Madrid, AnnArbor, CambridgeUS, EastLansing, Pittsburgh, Rochester, Antibes, JuanLesPins, Montpellier, Berkeley, PaloAlto, Goettingen, Marburg, Hannover, Berlin, Ilmenau, Dresden, Darmstadt, Frankfurt, Singapore, Szeged, Copenhagen, Luebeck, Malmoe, Warsaw
<code>\usecolortheme</code>	default, structure, sidebartab, albatross, beetle, crane, dove, fly, monarcha, seagull, wolverine, beaver, spruce, lily, orchid, rose, whale, seahorse, dolphin, seahorse
<code>\usefonttheme</code>	default, serif, structurebold, structureitalicserif, structuresmallcapsserif
<code>\useinnertheme</code>	default, circles, rectangles, rounded, inmargin
<code>\useoutertheme</code>	default, infolines, miniframes, smoothbars, sidebar, split, shadow, tree, smoothtree

Taulukko 5.1: `beamer`-luokan teemanvalintakomennot ja valmiita teemoja.

```
\usetheme{Bergen}
\usecolortheme{albatross}
\usefonttheme{serif}
```

Teemanvalintakomennoille voi antaa hakasulkeissa valinnaisen argumentin, jonka avulla määritetään kyseisen teeman asetuksia, jos teema sellaisia tukee. Asetukset ovat teemakohtaisia, ja niistä voi lukea lisää dokumenttiluokan ohjekirjasta. Seuraavassa on yleisesimerkki komentojen rakenteesta:

```
\usetheme[valitsin]{teeman nimi}
\useoutertheme[valitsin=arvo]{teeman nimi}
```

Fonttiteema `serif` vaihtaa koko diaesityksen kirjainperheeksi antiikvan. Sen myötä oletuksena olevaa groteskia (`sans serif`) ei siis käytetä enää missään. `serif`-teema sisältää kuitenkin muutaman valitsimen, joilla tähän voi tehdä pieniä hyödyllisiä poikkeuksia. Valitsimia on koottu taulukkoon 5.2, ja niitä voi antaa useampia kerralla.

Seuraava esimerkki asettaa groteskin suuriin elementteihin eli otsikoihin ja pieniin elementteihin eli dian reunojen mahdollisiin lisätietoihin. Muualla käytetään antiikvaa.

Valitsin	Merkitys
stillsansseriflarge	suuret elementit eli otsikot käyttävät groteskifonttia
stillsansserifsmall	pienet elementit kuten dian reuna-alueiden lisätiedot käyttävät groteskia
stillsansseriftext	normaali teksti groteskilla
stillsansserifmath	matematiikkatilassa groteski

Taulukko 5.2: `beamer`-luokan serif-fonttiteeman asetusvalitsimia.

```
\usefonttheme[stillsansseriflarge, stillsansserifsmall]{serif}
```

5.1.3 Muita asetuksia

Jos haluaa vaikuttaa `beamer`-dokumenttien otsikoiden tai muiden rakenteellisten osien fontteihin ja väreihin, täytyy käyttää dokumentti-luokan omia komentoja. Niiden käyttöä neuvotaan tässä alaluvussa. Sen sijaan diojen sisällä väliaikaiset kirjaintyyppin tai -leikkauksen muutokset tehdään samalla tavalla kuin Latexissa muutenkin. Fonttien tekniikkaa käsitellään luvussa 3.4 ja tekstin korostamisen typografiaa luvussa 4.2.

`beamer`-luokan erityisten tekstielementtien kuten otsikoiden fonttiin voi vaikuttaa komennolla `\setbeamerfont`. Sen ensimmäinen argumentti on tekstielementin nimi ja toinen argumentti sisältää fonttiasetukset. Esimerkiksi seuraava komento muuttaa aloitusdian (`\maketitle`) otsikon kirjainperheen ja -leikkauksen:

```
\setbeamerfont{title}{family=\rmfamily, series=\bfseries,
  shape=\itshape, size=\Huge}
```

Aloitusdian otsikko on tekstielementti nimeltä `title`, ja siksi se oli edellisen komennon ensimmäisenä argumenttina. Muita oleellisia tekstielementtejä on koottu taulukkoon 5.3.

Textsielementtien väreihin vaikutetaan komennolla `\setbeamercolor`, joka toimii lähes samalla tavalla kuin edellä esitelty komento `\setbeamerfont`. Värin asettamisessa käytetään valitsimia `fg` ja `bg`, joista ensin mainittu vaihtaa varsinaisen värin (foreground) ja jälkimmäinen taustavärin (background). Seuraavassa esimerkissä vaihdetaan diojen otsikon teksti valkeaksi ja tausta siniseksi:

Tekstielementti	Merkitys
title	aloitUSDian otsikko (<code>\maketitle</code>)
frametitle	diojen otsikot
framesubtitle	diojen alaotsikot
blocktitle	block-ympäristön otsikot
normal text	normaali teksti diojen sisällä
footnote	alaviitteet
item	luetelmien luettelomamerkit

Taulukko 5.3: `beamer`-luokan tekstielementtejä. Elementtien nimiä käytetään esimerkiksi komentojen `\setbeamerfont` ja `\setbeamercolor` kanssa.

```
\setbeamercolor{frametitle}{fg=white, bg=blue}
```

Värien nimien täytyy olla ennalta määriteltyjä. Perusvärit (`white`, `blue`, `red` ym.) on määritelty valmiiksi, mutta lisää värejä voi määrittellä ohjeilla, joita kerrotaan luvussa 4.2.7.

Oletuksena luettelmaympäristöt `itemize` ja `enumerate` latovat luettelomamerkit eri värillä kuin normaalin tekstin. Väri riippuu käytetystä väriteemasta (`\usecolortheme`). Jos haluaa, että luettelomamerkit ovat samalla värillä kuin normaali teksti, kannattaa käyttää seuraavan esimerkin komentoa:

```
\setbeamercolor{item}{parent={normal text}}
```

Edellisessä esimerkissä viitataan `item`-nimiseen tekstielementtiin, joka tarkoittaa luettelomamerkkejä. Valitsin `parent` tarkoittaa, että väri peritään toiselta tekstielementiltä, tässä tapauksessa normaalilta tekstiltä (`normal text`). Toki tässäkin voi käyttää valitsimia `fg` ja `bg`.

Normaalin tekstin taustaväri tarkoittaa koko dian tekstialueen taustaväriä. Seuraavassa on käytännön esimerkki, joka muuttaa diojen otsikon taustan kirkkaan vihreäksi (70 %) ja tekstialueen taustan vaalean vihreäksi (30 %).

```
\setbeamercolor{frametitle} {fg=black, bg=green!90}
\setbeamercolor{normal text}{fg=black, bg=green!30}
```

Numeroimattomien luetelmien eli `itemize`-ympäristön luettelomamerkki on `beamer`-dokumenttiluokassa oletuksena kolmionmuotoinen, mutta

asetusta voi muuttaa komennolla `\setbeamertemplate`. Tällä komennolla tehdään sekalaisia diojen asetuksia, joista annetaan tässä yhteydessä vain pari esimerkkiä. Luettelamerkki vaihdetaan seuraavan esimerkin komennolla.

```
\setbeamertemplate{itemize item}[circle]           % perustaso
\setbeamertemplate{itemize subitem}[square]         % toinen taso
\setbeamertemplate{itemize subsubitem}[triangle]   % kolmas taso
```

Vielä monipuolisemmin voi luettelamerkkeihin vaikuttaa, kun vaihtaa hakasulkeissa olevan argumentin tilalle aaltosulkeet. Tällaiseen argumenttiin voi kirjoittaa suunnilleen mitä hyvänsä Latex-komentoja, joilla luettelamerkki tuotetaan. Seuraavassa esimerkissä tätä voimakasta komentomuotoa käytetään maltillisesti pelkästään ajatusviivan tuottamiseen.

```
\setbeamertemplate{itemize item}{--}
```

`\setbeamertemplate`-komennolla voi asettaa myös diojen oikeaan alareunaan ladottaviin navigointisymboleihin. Tässä yhteydessä ei käsitellä niitä sen syvällisemmin, mutta navigointisymbolien poistaminen onnistuu helposti seuraavalla komennolla:

```
\setbeamertemplate{navigation symbols}{} 
```

5.2 Kirjeet

Latexin `letter`-dokumenttiluokka on tarkoitettu kirjeiden latomiseen. Tyyllisesti se soveltuu ehkä paremmin virallisiin kirjeisiin kuin henkilökohtaisiin. Esimerkiksi joukkojakelukirjeiden tekemiseen se voi olla käytännöllinen, koska tietokoneohjelman avulla voi helposti tuottaa Latex-koodia eli lähes samanlaisia kirjeitä eri vastaanottajille.

Esimerkissä 5.2 on yhden kirjeen runko ja oleelliset komennot. Dokumentin esittelyosassa olevalla `\address`-komennolla määritellään lähettäjän nimi ja osoitetiedot. Ne ladotaan jokaisen kirjeen alkuun. Myös komennolla `\signature` ilmaistaan lähettäjän nimi eli allekirjoitus, joka ladotaan kirjeen loppuun. Komento `\makelabels` aiheuttaa sen, että kaikkien kirjeiden jälkeen ladotaan sivu (tai useampia), jossa ovat

```

1 \documentclass{letter}
2 \usepackage[a4paper]{geometry}
3 \usepackage{polyglossia}
4 \setdefaultlanguage{finnish}
5
6 \address{Liisa Lähettäjä \\ Katuosoite 1 \\ 00000 Kaupunki}
7 \signature{Liisa Lähettäjä}
8
9 \makelabels
10
11 \begin{document}
12
13 \begin{letter}{Virpi Vastaanottaja \\ Tiennimi 3 \\ 99999 Kunta}
14
15   \opening{Hei!}
16
17   Tässä kirjeessä ei ole kovin mielenkiintoista sisältöä, mutta tähän
18   kohtaan se kirjoitettaisiin.
19
20   \closing{Terveisin}
21
22   \ps{Jk. Tässä on kirjeen jälkikirjoitus.}
23
24   \cc{Mauno Muuhenkilö}
25   \encl{lippu, lappu, paperi}
26
27 \end{letter}
28
29 \end{document}

```

Esimerkki 5.2: Latexin `letter`-dokumenttiluokka on tarkoitettu kirjeiden latomiseen.

kaikkien vastaanottajien osoitetiedot. Tämä on tarkoitettu osoitetarrojen tulostamiseen.

Varsinainen kirjeen sisältö toteutetaan `letter`-ympäristön avulla. Ympäristölle annetaan yksi argumentti, jossa on kyseisen kirjeen vastaanottajan nimi ja osoitetiedot. Itse kirje alkaa `\opening`-komennolla, jolla ilmaistaan tervehdys tai muut kirjeen aloitussanat.

Kirjeen lopussa `\closing`-komennolla on sopivaa ilmaista lopputoivotus tai muu vastaava. Lähettäjän nimi ladotaan sen jälkeen automaattisesti, jos lähettäjä on ilmaistu aiemmin `\signature`-komennolla. Mahdollisen jälkikirjoituksen voi ilmaista komennolla `\ps`, kirjeen jakelutietoja komennolla `\cc` ja liitteet komennolla `\encl`.

Yksi Latex-lähdedokumentti voi sisältää useita `letter`-ympäristöjä, ja jokainen niistä muodostaa erillisen, uudelta sivulta alkavan kirjeen. Kaikkiin kirjeisiin ladotaan sama lähettäjä, ellei lähettäjä tietoja välillä vaihda `\address`- ja `\signature`-komennolla.

Kirjallisuutta

- Berry, Karl – Gilmore, Stephen – Martinsen, Torsten (2018): *Latex 2ε: An unofficial reference manual*. October 2018. Saatavissa: <<https://latexref.xyz/>>.
- CTAN (1992–): *Comprehensive TeX Archive Network*. Saatavissa: <<https://www.ctan.org/>>.
- Itkonen, Markus (2015): *Kirjaintyyppit ja tyyli*. RPS-yhtiöt.
- Itkonen, Markus (2019): *Typografian käsikirja*. 5. painos. Typoteekki. Graafinen suunnittelu Markus Itkonen Oy.
- Kaijanaho, Antti-Juhani (1998): *Latex ja AMS-Latex. Opus asiatekstin ladonnasta*. Jyväskylän yliopiston atk-keskus. Saatavissa: <<http://urn.fi/URN:ISBN:978-951-39-7159-5>>.
- Kankaanpää, Salli – Heikkilä, Elina – Korhonen, Riitta – Maamies, Sari – Piehl, Aino – Eronen, Riitta – Hyvärinen, Riitta – Kolehmainen, Taru – Mikkonen, Pirjo – Moilanen, Raija – Pyhälähti, Minna – Räsänen, Matti – Vaula, Sari – Vihonen, Inkaliisa – Wihuri, Elina (2017): *Kielitoimiston oikeinkirjoitusopas*. 13. painos. Kotimaisten kielten keskuksen julkaisuja 177. Kotimaisten kielten keskus.
- Korpela, Jukka K. (2015): ”Tietokone typografian tuhoajana”. *Skrolli. Tietokonekulttuurin erikoislehti* 2015.3, s. 10–14. Saatavissa: <<https://www.skrolli.fi/numerot/2015-3/>>.
- Oetiker, Tobias – Partl, Hubert – Hyna, Irene – Schlegl, Elisabeth (2018): *The Not So Short Introduction to Latex 2ε. Or Latex 2ε in 139 minutes*. Version 6.2. Saatavissa: <<https://www.ctan.org/pkg/lshort-english>>.

Oetiker, Tobias – Partl, Hubert – Hyna, Irene – Schlegl, Elisabeth –
Hellgren, Timo (2005): *Pitkänpuoleinen johdanto Latex 2ε:n käyttöön.*
Eli opi Latex 2ε 133 minuutissa. Versio 4.17fi. Saatavissa: <[https://
www.ctan.org/pkg/lshort-finnish](https://www.ctan.org/pkg/lshort-finnish)>.

Asiahakemistot

Dokumenttiluokat

article, 18, 19, 52, 59–61, 86, 144–146, 152, 206, 220	leaflet, 65
beamer, 61, 64, 245–251	letter, 61, 252, 253
book, 60, 61, 63, 64, 75, 145, 152–156, 197, 206	memoir, 64
cd, 65	report, 60, 61, 144, 145, 152, 153, 155, 156, 206
guitartabs, 65	slides, 61, 64
	wallcalendar, 65

Komennot

Tekstitilan ja matematiikkatilan komennot on ladottu eri väreillä:
`\teksti` ja `\matematiikka`.

<code>\'</code> , 31, 178, 179	<code>\-</code> , 30, 103, 104, 109, 110, 125, 146, 178, 179
<code>\(</code> , 235	<code>\--</code> , 105, 106, 109
<code>\)</code> , 235	<code>\---</code> , 105
<code>\+</code> , 178, 179	<code>\-/</code> , 105, 106, 109
<code>\,</code> , 25, 26, 44, 119–121	<code>\.</code> , 31
<code>\,</code> , 241	<code>\:</code> , 241

`\;`, 241
`\<`, 178, 179
`\=`, 31, 177–179
`\=/`, 105–107
`\==`, 105, 106
`\===`, 105
`\>`, 177–179
`\[`, 34, 235, 237
`\#`, 22
`\$`, 22, 33
`\%`, 22
`\&`, 22
`\^`, 22, 31, 33
`_`, 22, 33
`\|`, 23, 55, 122–124, 168–172, 174, 175, 177
`\|`, 240
`\|*`, 123
`\"`, 31
`\{`, 22, 33, 34
`\}`, 22, 33, 34
`\~`, 22, 31, 33
`\]`, 34, 235, 237
`\``, 27, 31, 178, 179

`\a'`, 179
`\a=`, 179
`\a``, 179
`\acute`, 241
`\addbibresource`, 212
`\addcontentsline`, 147, 156, 185
`\addfontfeatures`, 27, 28, 87, 92, 94–96, 134
`\address`, 252–254

`\addto`, 106, 107, 151, 155, 182, 185, 206, 225
`\addtocounter`, 48, 50–52
`\addtolength`, 44
`\addvspace`, 117, 156, 157, 185
`\AE`, 31
`\ae`, 31
`\aleph`, 243
`\Alph`, 49, 160
`\alph`, 49, 51, 160
`\Alpha`, 241
`\alpha`, 241
`\alsoname`, 225
`\amalg`, 243
`\and`, 143
`\angle`, 243
`\appendix`, 152, 154
`\appendixname`, 152
`\approx`, 237, 242
`\arabic`, 49, 51, 148, 149, 160, 162, 166, 193, 196, 236
`\arraybackslash`, 175
`\arraystretch`, 169, 170, 181
`\ast`, 243
`\asymp`, 242
`\AtBeginBibliography`, 217
`\author`, 143, 144, 246

`\b`, 31
`\babelfont`, 101, 102
`\babeltags`, 101
`\backmatter`, 60, 152, 153
`\backslash`, 243
`\bar`, 241
`\baselinestretch`, 83, 89

`\begin`, 19, 40–42, 161, 176
`\Beta`, 241
`\beta`, 241
`\bfseries`, 84, 85, 89, 134, 149, 150, 156, 157, 160, 166, 175
`\bibfont`, 206, 215
`\bibitem`, 204, 205
`\bibname`, 206
`\bibsection`, 206
`\bigbreak`, 116
`\bigcap`, 242
`\bigcirc`, 243
`\bigcup`, 242
`\bigodot`, 242
`\bigoplus`, 242
`\bigotimes`, 242
`\bigskip`, 116, 228
`\bigsqcup`, 242
`\bigtriangledown`, 243
`\bigtriangleup`, 243
`\biguplus`, 242
`\bigvee`, 242
`\bigwedge`, 242
`\bot`, 243
`\bottomfraction`, 188
`\bottomrule`, 176
`\bowtie`, 242
`\breve`, 241
`\bullet`, 243

`\c`, 31
`\cap`, 242
`\caption`, 38, 181, 182, 184, 190, 193, 194
`\captionsetup`, 191, 192

`\captionsfinnish`, 106, 107, 151, 155, 182, 185, 206, 225
`\cc`, 253, 254
`\cdot`, 238, 242
`\cdots`, 243
`\Centering`, 115
`\centering`, 115, 175
`\chapter`, 60, 63, 71, 145, 148, 151, 152, 154, 155, 188, 193, 197, 227, 247
`\chapter*`, 147, 227
`\chaptermark`, 74, 75
`\chaptername`, 75, 151
`\check`, 241
`\Chi`, 241
`\chi`, 241
`\circ`, 243
`\cite`, 213, 216
`\citealt`, 205
`\citealt*`, 205
`\citeauthor`, 205, 213
`\citeauthor*`, 205
`\citep`, 205
`\citep*`, 205
`\citet`, 205
`\citet*`, 204, 205
`\citetitle`, 213
`\citeyear`, 205, 213
`\citeyearpar`, 205
`\cleardoublepage`, 141, 145, 154, 188
`\clearpage`, 20, 141, 145, 151, 188, 228
`\cline`, 169, 170, 174
`\closing`, 253, 254

`\clubpenalty`, 125
`\clubsuit`, 243
`\cmidrule`, 176
`\color`, 140
`\colorbox`, 140
`\columnbreak`, 202
`\cong`, 242
`\contentslabel`, 156, 157, 185
`\contentsname`, 155
`\contentspage`, 156, 157, 185
`\ContinuedFloat`, 192, 193
`\coprod`, 242
`\copyright`, 33
`\counterwithin`, 50
`\counterwithin*`, 51, 197
`\counterwithout`, 50
`\counterwithout*`, 51
`\cup`, 242

`\d`, 31
`\dag`, 33
`\dagger`, 243
`\dashuline`, 133
`\dashv`, 242
`\date`, 143, 144, 246
`\dblfloatpagefraction`, 188
`\dbltopfraction`, 188, 202
`\ddag`, 33
`\ddagger`, 243
`\ddot`, 241
`\ddots`, 243
`\DeclareBibliographyDriver`, 219
`\DeclareDelimFormat`, 215–219
`\DeclareFieldFormat`, 217–219
`\DeclareFloatFont`, 191

`\DeclareNameAlias`, 215
`\DeclareNewFloatType`, 183
`\DeclareSortingTemplate`, 219
`\defaultfontfeatures`, 91
`\DefineBibliographyStrings`, 219
`\definecolor`, 139, 140
`\Delta`, 241
`\delta`, 241
`\DH`, 31
`\dh`, 31
`\diamond`, 243
`\diamondsuit`, 243
`\disablehyphenation`, 110
`\discretionary`, 107–110
`\div`, 242
`\DJ`, 31
`\dj`, 31
`\documentclass`, 19, 52, 59, 75, 86, 88, 154, 220, 246, 253

`\dot`, 241
`\doteq`, 242
`\dotfill`, 133
`\dots`, 29, 33
`\dots`, 243
`\dotuline`, 133
`\Downarrow`, 243
`\downarrow`, 243
`\draw`, 231–234

`\ell`, 243
`\emph`, 130, 132
`\emptyset`, 243
`\enablehyphenation`, 110
`\encl`, 253, 254
`\end`, 19, 40–42, 119, 176, 177

`\Endash`, 105
`\endtabularx`, 176
`\englishfont`, 100
`\englishfontsf`, 100
`\englishfonttt`, 100
`\enlargethispage`, 142
`\enlargethispage*`, 143
`\enquote`, 27, 28
`\enspace`, 44
`\Epsilon`, 241
`\epsilon`, 241
`\equiv`, 242
`\Eta`, 241
`\eta`, 241
`\exists`, 243
`\extracolsep`, 173

`\fancyhf`, 72–75
`\fancypagestyle`, 73–75
`\fbox`, 53, 54
`\fcolorbox`, 140
`\figurename`, 182
`\flat`, 243
`\FloatBarrier`, 188
`\floatbox`, 189, 190
`\floatpagefraction`, 188
`\floatsetup`, 190, 191
`\flushbottom`, 142
`\flushcolumns`, 202
`\fnsymbol`, 49, 196
`\fontsize`, 82–84, 87–89
`\footcite`, 213
`\footnote`, 195, 196, 198
`\footnotemark`, 196
`\footnotesize`, 85, 88, 196

`\footnotetext`, 196
`\footrulewidth`, 74, 75
`\forall`, 243
`\foreignlanguage`, 101
`\frac`, 235, 238, 239
`\framebox`, 53, 54
`\frontmatter`, 60, 152, 153
`\frown`, 242
`\fussy`, 114

`\Gamma`, 241
`\gamma`, 241
`\ge`, 242
`\geometry`, 65–67, 70, 197
`\geq`, 242
`\gets`, 243
`\gg`, 242
`\grave`, 241
`\greekfont`, 100
`\greekfontsf`, 100
`\greekfonttt`, 100
`\guillemotleft`, 32
`\guillemotright`, 32
`\guilsinglleft`, 32
`\guilsinglright`, 32

`\H`, 31
`\hangpara`, 119, 120
`\hat`, 241
`\hbar`, 243
`\headrulewidth`, 74, 75
`\heartsuit`, 243
`\hfill`, 47, 162, 166
`\hline`, 169, 170, 172, 174, 181
`\hookleftarrow`, 243
`\hookrightarrow`, 243

`\href`, 137, 138
`\hrulefill`, 133
`\hspace`, 43, 44, 46, 47, 49, 150, 157, 169–171, 177–179
`\hspace*`, 44
`\Huge`, 85
`\huge`, 85
`\hypersetup`, 76, 137
`\Hyphdash`, 105
`\hyphenation`, 103

`\i`, 31
`\iff`, 243
`\ifthenelse`, 40
`\ignorespaces`, 39, 42, 120, 180
`\ignorespacesafterend`, 42
`\Im`, 243
`\imath`, 243
`\in`, 242
`\include`, 20
`\includegraphics`, 229, 230
`\indent`, 118
`\index`, 35, 220, 222–226
`\indexname`, 221
`\indexprologue`, 228
`\indexsetup`, 227
`\infty`, 238, 243
`\input`, 20
`\institute`, 246, 247
`\int`, 238, 242
`\Iota`, 241
`\iota`, 241
`\item`, 121–123, 158–160, 162, 163, 246
`\itshape`, 25, 84, 85, 96, 149

`\j`, 31
`\jmath`, 243
`\justifying`, 115

`\k`, 31
`\Kappa`, 241
`\kappa`, 241
`\kill`, 177–179

`\L`, 31
`\l`, 31
`\label`, 50, 182, 190, 194, 195, 236
`\labelenumi`, 160
`\labelenumii`, 160
`\labelenumiii`, 160
`\labelenumiv`, 160
`\labelitemi`, 160
`\labelitemii`, 160
`\labelitemiii`, 160
`\labelitemiv`, 160
`\Lambda`, 241
`\lambda`, 241
`\langle`, 243
`\LARGE`, 85
`\Large`, 85, 88, 89, 149, 150
`\large`, 85, 87, 88, 134, 149, 156, 157, 166
`\LaTeX`, 31
`\lceil`, 243
`\ldots`, 29, 33
`\le`, 242
`\left`, 235, 238
`\Leftarrow`, 243
`\leftarrow`, 243
`\leftharpoondown`, 243
`\leftharpoonup`, 243

`\lefthyphenmin`, 106
`\leftmark`, 71, 74, 75
`\Leftrightarrow`, 243
`\leftrightharrow`, 243
`\leq`, 242
`\lettrine`, 128
`\lfloor`, 243
`\lgroup`, 243
`\lim`, 238, 242
`\linespread`, 19, 83, 84, 86, 88, 89, 122, 123
`\listfigurename`, 185
`\listof`, 184
`\listoffigures`, 184
`\listoftables`, 184
`\listtablename`, 185
`\ll`, 242
`\lmoustache`, 243
`\lnot`, 243
`\Longleftarrow`, 243
`\longleftarrow`, 243
`\Longleftrightharrow`, 243
`\longleftrightharrow`, 243
`\longmapsto`, 243
`\Longrightarrow`, 243
`\longrightarrow`, 243
`\lq`, 32

`\mainmatter`, 60, 152, 153
`\makebox`, 53, 56, 57, 120, 121, 179, 180
`\makeindex`, 220, 221, 223, 227, 228
`\makelabel`, 162, 163, 165–167
`\makelabels`, 252, 253
`\maketitle`, 143, 246, 247, 250, 251

`\mapsto`, 243
`\marginpar`, 125, 126
`\markboth`, 71, 72, 74, 75
`\markright`, 72, 74
`\mathbf`, 240
`\mathcal`, 240
`\mathit`, 240
`\mathring`, 241
`\mathrm`, 240
`\mathsf`, 240
`\mathtt`, 240
`\mbox`, 53, 104
`\mdseries`, 84
`\medbreak`, 116
`\medskip`, 116
`\mid`, 242
`\midrule`, 176
`\mkbibnamefamily`, 217
`\mkbibnamegiven`, 217
`\mkbibnameprefix`, 217
`\mkbibnamesuffix`, 217
`\mkbibparens`, 218
`\mkbibquote`, 218
`\models`, 242
`\mp`, 242
`\Mu`, 241
`\mu`, 241
`\multicolumn`, 169, 171, 172
`\multirow`, 171, 173, 174

`\nabla`, 243
`\nameref`, 195
`\natural`, 243
`\ne`, 242
`\nearrow`, 243

`\neg`, 243
`\neq`, 242
`\newcolumn`type, 175
`\newcommand`, 35–39, 120, 126, 138, 151, 180, 226, 233
`\newcommand*`, 37, 38
`\newcounter`, 48–51, 162
`\newenvironment`, 40–42, 166, 175, 176
`\newenvironment*`, 42
`\newfontface`, 81
`\newfontfamily`, 81, 88, 91, 100
`\newgeometry`, 66
`\newlength`, 44, 45, 120, 180
`\newline`, 124, 172
`\newpage`, 52, 141
`\NG`, 31
`\ng`, 31
`\ni`, 242
`\nocite`, 213
`\noclub`, 125
`\node`, 233, 234
`\noindent`, 118, 167, 180, 201
`\NoIndentAfterEnv`, 118
`\nolinkurl`, 138
`\nopagebreak`, 142
`\nopagecolor`, 140
`\normalfont`, 160
`\normalmarginpar`, 127
`\normalsize`, 85, 87–89, 149, 157
`\not`, 242
`\note`, 64
`\notin`, 242
`\nowidow`, 125
`\Nu`, 241
`\nu`, 241
`\narrow`, 243
`\O`, 31
`\o`, 31
`\obeycr`, 23
`\odot`, 242
`\OE`, 31
`\oe`, 31
`\oint`, 242
`\Omega`, 241
`\omega`, 241
`\ominus`, 242
`\onecolumn`, 199
`\opening`, 253, 254
`\oplus`, 242
`\oslash`, 242
`\otimes`, 242
`\overbrace`, 239, 242
`\overleftarrow`, 239, 242
`\overleftrightharrow`, 242
`\overline`, 242
`\overrightarrow`, 239, 242
`\P`, 33
`\pagebreak`, 142
`\pagecolor`, 140
`\pagenumbering`, 72
`\pageref`, 51, 52, 194, 195
`\pagestyle`, 70–72, 75, 154–156
`\par`, 24, 113, 120
`\paragraph`, 145, 146
`\paragraph*`, 147
`\parallel`, 242
`\parbox`, 54–56
`\parencite`, 213, 216

`\part`, 145, 146, 148, 151, 155, 188
`\part*`, 147
`\partname`, 151
`\path`, 232
`\pause`, 246, 247
`\pdfbookmark`, 77, 144, 156
`\perp`, 242
`\phantomsection`, 147
`\Phi`, 241
`\phi`, 241
`\Pi`, 241
`\pi`, 237, 241
`\pm`, 242
`\poptabs`, 178
`\pounds`, 33
`\prec`, 242
`\preceq`, 242
`\prime`, 243
`\printbibliography`, 154, 212
`\printindex`, 154, 220, 221, 227, 228
`\prod`, 242
`\propto`, 242
`\protect`, 38, 146
`\providecommand`, 36
`\providecommand*`, 37
`\ps`, 253, 254
`\Psi`, 241
`\psi`, 241
`\pushtabs`, 178

`\qqquad`, 238, 240
`\quad`, 44
`\quad`, 239, 240
`\quotedblbase`, 32

`\quotesinglbase`, 32

`\r`, 31
`\raggedbottom`, 142, 198
`\raggedcolumns`, 202
`\RaggedLeft`, 115
`\raggedleft`, 115, 175
`\RaggedRight`, 115, 126
`\raggedright`, 111, 115, 150, 175
`\raisebox`, 56
`\rangle`, 243
`\rceil`, 243
`\Re`, 243
`\ref`, 50–52, 194, 195, 236
`\refname`, 206
`\refstepcounter`, 50
`\regtotcounter`, 51, 52
`\renewcommand`, 36, 37, 39, 51, 74, 75, 88, 133, 148, 149, 151, 155, 156, 160, 162, 163, 166, 169, 170, 181, 188, 193, 196, 206, 215, 225, 236
`\renewcommand*`, 37
`\renewenvironment`, 40, 41
`\renewenvironment*`, 42
`\resizebox`, 56, 57
`\resizebox*`, 57
`\restorecr`, 23
`\restoregeometry`, 66
`\reversemarginpar`, 127
`\rfloor`, 243
`\rgroup`, 243
`\Rho`, 241
`\rho`, 241

`\right`, 235, 238
`\Rightarrow`, 243
`\rightarrow`, 243
`\rightharpoondown`, 243
`\rightharpoonup`, 243
`\righthyphenmin`, 106
`\rightleftharpoons`, 243
`\rightmark`, 71, 74, 75
`\rmfamily`, 84, 91, 156, 157, 185
`\rmoustache`, 243
`\Roman`, 49
`\roman`, 49, 160
`\rotatebox`, 56, 57
`\rowcolors`, 176
`\rq`, 32
`\rule`, 31, 32, 133

`\S`, 33
`\scalebox`, 56
`\scriptsize`, 85, 126
`\scshape`, 84, 94
`\searrow`, 243
`\section`, 38, 60, 71, 145–150, 189, 194, 201, 206, 227, 236, 246, 247
`\section*`, 147, 227
`\sectionbreak`, 151
`\sectionmark`, 74, 75
`\see`, 224
`\seealso`, 224
`\seename`, 225
`\selectfont`, 82–84, 87
`\selectlanguage`, 101
`\setbeamercolor`, 250, 251
`\setbeamerfont`, 250, 251
`\setbeamertemplate`, 252
`\setcitestyle`, 203, 204
`\setcounter`, 48, 49, 148, 154, 155, 160, 166, 187, 206
`\setdefaultlanguage`, 19, 28, 99, 246
`\setlength`, 23, 44, 46, 54, 68, 114–116, 118, 120–122, 127, 133, 140, 160–163, 166, 170, 179, 180, 187, 198, 200, 202, 206, 214
`\setmainfont`, 19, 79–81, 86, 88, 90, 95, 96, 101
`\setmathfont`, 96, 98
`\setminus`, 242
`\setmonofont`, 79, 80, 86, 88, 90, 91, 96, 101
`\setnoclub`, 124
`\setnowidow`, 124
`\setotherlanguage`, 99
`\setsansfont`, 79, 80, 86, 88, 90, 96, 101
`\settodepth`, 45
`\settoheight`, 45
`\settowidth`, 45
`\sffamily`, 84, 89, 91, 126, 134, 149, 150
`\sharp`, 243
`\Sigma`, 241
`\sigma`, 241
`\signature`, 252–254
`\sim`, 242
`\simeq`, 242
`\sloppy`, 114
`\slshape`, 84, 85, 96

`\small`, 85, 87, 88, 122, 123, 157, 185
`\smallbreak`, 116
`\smallskip`, 116
`\smile`, 242
`\sout`, 133
`\spadesuit`, 243
`\sqcap`, 242
`\sqcup`, 242
`\sqrt`, 239
`\sqsubset`, 242
`\sqsubseteq`, 242
`\sqsupset`, 242
`\sqsupseteq`, 242
`\SS`, 31
`\ss`, 31
`\star`, 243
`\stepcounter`, 50
`\stretch`, 47
`\strut`, 32, 54
`\subparagraph`, 145, 146
`\subparagraph*`, 147
`\subsection`, 60, 145, 146, 148, 149, 194, 247
`\subsection*`, 147
`\subsectionmark`, 74
`\subset`, 242
`\subseteq`, 242
`\subsubsection`, 145, 146, 149
`\subsubsection*`, 147
`\succ`, 242
`\succeq`, 242
`\sum`, 238, 242
`\supset`, 242
`\supseteq`, 242
`\surd`, 243
`\swarrow`, 243
`\tablename`, 182
`\tableofcontents`, 153, 154, 156
`\tabularx`, 176
`\Tau`, 241
`\tau`, 241
`\TeX`, 31
`\textasciicircum`, 22, 33
`\textasciitilde`, 22, 33
`\textasteriskcentered`, 33, 160
`\textbackslash`, 22, 33, 35, 223
`\textbar`, 33
`\textbardbl`, 33
`\textbf`, 84, 129
`\textbigcircle`, 33
`\textbraceleft`, 22, 33
`\textbraceright`, 22, 33
`\textbullet`, 33, 160, 161, 166, 223
`\textcite`, 213, 216, 219
`\textcolor`, 140
`\textcompwordmark`, 92
`\textcopyright`, 33
`\textdagger`, 33
`\textdaggerdbl`, 33
`\textdollar`, 22, 33
`\textellipsis`, 29, 33
`\textemdash`, 33
`\textendash`, 33, 160
`\textenglish`, 35, 99
`\texteuro`, 33
`\textexclamdown`, 33
`\textfraction`, 188
`\textgreater`, 33

`\textgreek`, 99
`\textit`, 84, 129, 224
`\textleftarrow`, 33
`\textless`, 33
`\textmd`, 84
`\textordfeminine`, 33
`\textordmasculine`, 33
`\textparagraph`, 33
`\textperiodcentered`, 33, 160
`\textquestiondown`, 33
`\textquotedbl`, 27, 32
`\textquotedblleft`, 32
`\textquotedblright`, 32
`\textquoteleft`, 32
`\textquoteright`, 32
`\textquotesingle`, 27, 32
`\textregistered`, 33
`\textrightarrow`, 33, 226
`\textrm`, 84
`\textsc`, 84, 96, 130, 163
`\textsection`, 33
`\textsf`, 84
`\textsl`, 84, 129
`\textsterling`, 33
`\textsubscript`, 29, 30
`\textsubscript*`, 30
`\textsuperscript`, 29, 30, 198
`\textsuperscript*`, 30
`\texttrademark`, 33
`\texttt`, 27, 28, 35, 84, 110, 135,
223
`\textunderscore`, 22, 33
`\textup`, 84
`\textvisiblespace`, 33
`\TH`, 31
`\th`, 31
`\thanks`, 143
`\the`, 46
`\theauthor`, 144
`\thechapter`, 75, 193
`\thedate`, 144
`\theequation`, 236
`\thefigure`, 193
`\thefootnote`, 196
`\thempfootnote`, 197
`\thepage`, 49, 71–73, 75
`\thepart`, 148
`\thesection`, 75, 148–150, 236
`\thesubsection`, 148, 149
`\thesubsubsection`, 149
`\Theta`, 241
`\theta`, 241
`\thetable`, 49, 193
`\thetitle`, 144
`\thispagestyle`, 71, 155, 156
`\tilde`, 241
`\times`, 238, 242
`\tiny`, 85
`\title`, 143, 144, 246
`\titlecontents`, 156, 157, 184, 185
`\titleformat`, 149, 150
`\titleformat*`, 149
`\titlelabel`, 150
`\titlerule`, 157
`\titlerule*`, 157, 185
`\titlespacing`, 151
`\titlespacing*`, 150, 151
`\to`, 238, 243
`\today`, 143
`\top`, 243

`\topfraction`, 188, 202
`\toprule`, 176
`\total`, 51, 52
`\totvalue`, 51, 52
`\triangle`, 243
`\triangleleft`, 243
`\triangleright`, 243
`\ttfamily`, 84, 89, 91
`\twocolumn`, 199, 200

`\u`, 31
`\uline`, 133
`\ULthickness`, 133
`\underbrace`, 239, 242
`\underleftarrow`, 242
`\underleftrightharrow`, 242
`\underline`, 132
`\underline`, 242
`\underrightarrow`, 242
`\Uparrow`, 243
`\uparrow`, 243
`\Updownarrow`, 243
`\updownarrow`, 243
`\uplus`, 242
`\upshape`, 84
`\Upsilon`, 241
`\upsilon`, 241
`\url`, 138
`\urlstyle`, 137
`\usecolortheme`, 248, 249, 251
`\usecounter`, 162, 166
`\usefonttheme`, 248, 249
`\useinnertheme`, 248, 249
`\useoutertheme`, 248, 249

`\usepackage`, 18, 19, 28, 52, 65, 72, 75, 76, 79, 86, 88, 97, 99, 101, 105, 124, 132, 137, 141, 176, 189, 197–199, 203, 211, 216, 218, 220, 228, 246, 253
`\usetheme`, 248, 249
`\uuline`, 133
`\uwave`, 133

`\v`, 31
`\value`, 49
`\varepsilon`, 241
`\varphi`, 241
`\varpi`, 241
`\varrho`, 241
`\varsigma`, 241
`\vartheta`, 241
`\vdash`, 242
`\vdots`, 243
`\vec`, 239
`\vee`, 242
`\verb`, 135, 136
`\Vert`, 243
`\vert`, 243
`\vfill`, 47, 117
`\vline`, 171
`\vspace`, 43, 44, 47, 116, 117, 228
`\vspace*`, 44, 116

`\wedge`, 242
`\widehat`, 241
`\widetilde`, 241
`\widowpenalty`, 125
`\wp`, 243
`\wr`, 243

`\Xi`, 241
`\xi`, 241
`\xout`, 133

`\Zeta`, 241
`\zeta`, 241

Laskurit

`bottomnumber`, 187
`chapter`, 48, 52, 148, 152, 184
`dbltopnumber`, 187, 202
`enumi`, 48, 159, 160, 162, 166
`enumii`, 48, 159, 160
`enumiii`, 48, 159, 160
`enumiv`, 48, 159, 160
`equation`, 48, 236
`figure`, 48, 193
`footnote`, 48, 196, 197
`mpfootnote`, 48, 197

`page`, 48, 50–52, 71, 144, 197
`paragraph`, 48, 148
`part`, 48, 148
`secnumdepth`, 148, 149, 153, 154, 206
`section`, 48, 148, 152, 184
`subparagraph`, 48, 148
`subsection`, 48, 148, 149
`subsubsection`, 48, 148, 149
`table`, 48, 49, 193
`tocdepth`, 154, 155
`topnumber`, 187, 202
`totalnumber`, 187

Mitat

`\arrayrulewidth`, 170
`\baselineskip`, 32, 54, 82, 142
`\bibhang`, 206, 214
`\bibinitsep`, 214
`\bibitemsep`, 214
`\bibnamesep`, 214
`\bibsep`, 206
`\bigskipamount`, 116
`\columnsep`, 68, 200, 202
`\columnseprule`, 200, 202

`\columnwidth`, 69, 200
`\dblfloatsep`, 187, 202
`\dbltextfloatsep`, 187
`\depth`, 53, 56, 57
`\doublerulesep`, 170
`\emergencystretch`, 23, 114
`\fboxrule`, 54, 140
`\fboxsep`, 54, 140
`\FBwidth`, 189, 190

`\fill`, 47, 173
`\floatsep`, 187, 202
`\footnotemargin`, 198

`\height`, 53, 56, 57

`\intextsep`, 187
`\itemindent`, 121, 122, 163–165
`\itemsep`, 161–166

`\labelsep`, 163–166
`\labelwidth`, 163–165
`\leftmargin`, 121, 122, 162–166
`\leftmargini`, 160
`\leftmarginii`, 160
`\leftmarginiii`, 160
`\leftmarginiv`, 160
`\linewidth`, 69, 174–176, 200
`\listparindent`, 122, 164, 165

`\marginparpush`, 127
`\marginparsep`, 126, 127
`\marginparwidth`, 126, 127
`\medskipamount`, 116
`\multicolsep`, 202

`\paperheight`, 69
`\paperwidth`, 69
`\parindent`, 46, 118, 122, 167, 206,
214
`\parsep`, 122, 161, 162, 164–166
`\parskip`, 46, 115, 118, 161, 163,
164, 167
`\partopsep`, 122, 163–165

`\rightmargin`, 122, 164, 165

`\smallskipamount`, 116
`\spaceskip`, 23

`\tabbingsep`, 178, 179
`\tabcolsep`, 170
`\textfloatsep`, 187
`\textheight`, 69, 142
`\textwidth`, 44, 69, 200
`\topsep`, 122, 163, 164
`\totalheight`, 53, 56, 57

`\ULdepth`, 133

`\width`, 53, 54, 56, 57

Paketit

`amsmath`, 237, 240, 241

`babel`, 22, 28, 98, 100, 101, 104, 106,
107, 151, 155, 182, 206,
225
`biblatex`, 203, 207, 208, 210–213,
215–219
`biblatex-ext`, 212

`booktabs`, 176

`caption`, 189, 191, 192
`chngcntr`, 50, 51, 197
`color`, 138
`csquotes`, 27, 28, 210, 211

`extdash`, 105–107, 109

fancyhdr, 65, 72, 73, 75, 228
 fancyvrb, 136
 floatrow, 180, 183, 184, 186, 189
 fontspec, 18, 19, 27, 79, 80, 86–88,
 92, 96, 98, 100–102
 footmisc, 197–199

 geometry, 18, 19, 62, 65, 67, 68, 70,
 71, 126, 197, 200
 graphicx, 56, 62, 229

 hanging, 119
 hyperref, 62, 76, 77, 137, 144–147,
 153, 156, 195, 198, 199,
 246

 icomma, 237
 ifthen, 40
 imakeidx, 17, 220, 228
 indextools, 17, 220, 228

 latexsym, 241
 lettrine, 127, 128
 libertinus, 79, 97
 longtable, 176

 multicol, 64, 199–201, 222
 multirow, 171, 173, 174

 natbib, 203, 205, 207

 noindentafter, 118, 119
 nowidow, 124, 125

 pdfscape, 62, 66
 pgf, 231
 placeins, 188, 189
 polyglossia, 19, 22, 28, 98, 99, 106,
 107, 110, 119, 151, 155,
 182, 206, 211, 225, 246

 ragged2e, 115, 191, 192
 realscripts, 29, 30

 showhyphens, 108
 soul, 133

 tabularx, 167, 174–176
 tabulary, 176
 tikz, 231, 235
 titlesec, 141, 149, 150
 titletoc, 156, 157, 184, 185
 titling, 144
 totcount, 51
 totpages, 51, 52

 ulem, 132, 133
 unicode-math, 96–98

 xcolor, 138, 139, 176

Ympäristöt

Tekstitilan ja matematiikkatilan ympäristöt on ladottu eri väreillä: teksti ja **matematiikka**.

abstract, 60, 144
align, 237
align*, 237
array, 236, 238, 240

bfseries, 85
block, 247, 251
Bmatrix, 240
bmatrix, 240

Center, 115
center, 115, 167, 181
columns, 248

description, 157–159, 161
displaymath, 235
document, 19, 40, 154, 220

enumerate, 157–161, 246, 247, 251
equation, 63, 236

figure, 181, 182, 185, 186, 190,
191, 193, 200
figure*, 182, 187, 200, 202
floatrow, 189, 190
FlushLeft, 115
flushleft, 115
FlushRight, 115
flushright, 115
footnotesize, 85
frame, 246, 247

hangparas, 120
Huge, 85
huge, 85

itemize, 157–161, 251
itshape, 85

justify, 115

landscape, 66
LARGE, 85
Large, 85
large, 85
letter, 253, 254
list, 118, 121–123, 158, 161–166
longtable, 176

math, 235
matrix, 240
mdseries, 85
minipage, 56, 196, 197
multicols, 199–202
multicols*, 201

normalsize, 85

otherlanguage, 101

pmatrix, 240

quotation, 121, 122
quote, 121, 122

rmfamily, 85
scriptsize, 85
scshape, 85
sffamily, 85
slshape, 85
small, 85

tabbing, 177–179
table, 181, 182, 185, 189–191, 193,
 200
table*, 182, 187, 200, 202
tabular, 118, 167–169, 171–174,
 176, 181, 190
tabular*, 167, 173, 174

tabularx, 167, 174–176
thebibliography, 204–206
tikzpicture, 231–235
tiny, 85
titlepage, 144
ttfamily, 85

upshape, 85

Verbatim, 136
verbatim, 136
verse, 121, 122
Vmatrix, 240
vmatrix, 240