

Ne najkrajši uvod v L^AT_EX 2_ε

oziroma L^AT_EX 2_ε v 130 minutah

Tobias Oetiker

Hubert Partl, Irene Hyna in Elisabeth Schlegl

Version 4.20, May 31, 2006

slovenski prevod in priredba

Bor Plestenjak

Verzija 4.20.1, 12. november 2006

Copyright ©1995-2005 Tobias Oetiker and Contributors. All rights reserved.

Copyright ©2006 Bor Plestenjak for the Slovene translation and adaptation.
All rights reserved.

This document is free; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This document is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this document; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

Avtorske pravice ©1995-2005 Tobias Oetiker in ostali, ki so prispevali k nastanku LShort. Vse pravice pridržane.

Avtorske pravice ©2006 Bor Plestenjak za slovenski prevod in priredbo. Vse pravice pridržane.

Ta dokument je prost; lahko ga prosto širite in/ali spreminjate pod pogoji navedenimi v splošni javni licenci GNU General Public License, ki jo je objavila Free Software Foundation; ali različica 2 ali pa katera koli poznejša različica (po vaši izbiri).

Ta dokument je na razpolago z upanjem, da bo uporaben, toda BREZ KAKRŠNEKOLI GARANCIJE; celo brez privzetega jamstva DA JE VREDEN UPORABE ali PRIMEREN ZA DOLOČEN NAMEN. Za podrobnosti poglejte GNU General Public License.

Poleg tega dokumenta bi morali prejeti tudi kopijo GNU General Public License; v nasprotnem primeru lahko pišete na Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

Zahvala

Večji del snovi v navodilih izhaja iz avstrijskega uvoda v L^AT_EX 2.09, ki so ga v nemščini napisali:

Hubert Partl <partl@mail.boku.ac.at>
Zentraler Informatikdienst der Universität für Bodenkultur Wien

Irene Hyna <Irene.Hyna@bmwf.ac.at>
Bundesministerium für Wissenschaft und Forschung Wien

Elisabeth Schlegl <noemail>
Graz

Nemško različico dokumenta, ki jo je za L^AT_EX 2_ε posodobil Jörg Knappen, lahko najdete na

CTAN:/tex-archive/info/lshort/german

Naslednji posamezniki so s številnimi popravki, predlogi in drugimi pripombami pomagali izboljšati ta navodila. Vložili so veliko truda, da so mi pomagali spraviti dokument v sedanjo obliko, zato bi se jim na tem mestu rad najlepše zahvalil.

Rosemary Bailey, Marc Bevand, Friedemann Brauer, Jan Busa,
Markus Brühwiler, Pietro Braione, David Carlisle, José Carlos Santos,
Neil Carter, Mike Chapman, Pierre Chardaire, Christopher Chin, Carl Cerecke,
Chris McCormack, Wim van Dam, Jan Dittberner, Michael John Downes,
Matthias Dreier, David Dureisseix, Elliot, Hans Ehrbar, Daniel Flipo, David Frey,
Hans Fugal, Robin Fairbairns, Jörg Fischer, Erik Frisk, Mic Milic Frederickx,
Frank, Kasper B. Graversen, Arlo Griffiths, Alexandre Guimond, Andy Goth,
Cyril Goutte, Greg Gamble, Frank Fischli, Morten Høgholm, Neil Hammond,
Rasmus Borup Hansen, Joseph Hilferty, Björn Hvittfeldt, Martien Hulsen,
Werner Icking, Jakob, Eric Jacoboni, Alan Jeffrey, Byron Jones, David Jones,
Johannes-Maria Kaltenbach, Michael Koundouros, Andrzej Kawalec,
Sander de Kievit, Alain Kessi, Christian Kern, Tobias Klauser, Jörg Knappen,
Kjetil Kjernsmo, Maik Lehradt, Rémi Letot, Flori Lambrechts, Axel Liljencrantz,
Johan Lundberg, Alexander Mai, Hendrik Maryns, Martin Maechler,
Aleksandar S Milosevic, Henrik Mitsch, Claus Malten, Kevin Van Maren,
Richard Nagy, Philipp Nagele, Lenimar Nunes de Andrade, Manuel Oetiker,
Urs Oswald, Martin Pfister, Demerson Andre Polli, Nikos Pothitos,
Maksym Polyakov Hubert Partl, John Reffling, Mike Ressler, Brian Ripley,
Young U. Ryu, Bernd Rosenlecher, Chris Rowley, Risto Saarelma,
Hanspeter Schmid, Craig Schlenter, Gilles Schintgen, Baron Schwartz,
Christopher Sawtell, Miles Spielberg, Geoffrey Swindale, Laszlo Szathmary,
Boris Tobotras, Josef Tkadlec, Scott Veirs, Didier Verna, Fabian Wernli,
Carl-Gustav Werner, David Woodhouse, Chris York, Fritz Zaucker, Rick Zaccone
in Mikhail Zotov.

Predgovor

L^AT_EX [1] je program za urejanje besedil, ki je zelo primeren za izdelavo znanstvenih in matematičnih besedil z visoko tiskarsko kvaliteto. Program je uporaben tudi za izdelavo vseh vrst drugih dokumentov, od preprostih pisem do celotnih knjig. L^AT_EX uporablja T_EX [2] za stavljenje besedila.

Ta kratka navodila opisujejo L^AT_EX 2_ε in bi morala zadoščati za večino primerov uporabe L^AT_EXa. Kompletan opis L^AT_EXa lahko najdete v [1, 3], podrobnejša slovenska navodila pa v [18].

Navodila so razdeljena na 6 poglavij:

Poglavje 1 opisuje osnovno strukturo L^AT_EX 2_ε dokumentov. Tu spoznamo tudi nekaj zgodovine L^AT_EXa. Na koncu poglavja boste imeli grobo predstavo o tem, kako deluje L^AT_EX.

Poglavje 2 obravnava podrobnosti sestavljanja dokumentov. Razloži večino ključnih ukazov v samem L^AT_EXu in v okolju, v katerem ga poganjamo. Po tem poglavju boste lahko napisali svoj prvi tekst v L^AT_EXu.

Poglavje 3 razloži, kako se v L^AT_EX vstavlja matematične formule. Veliko zgledov pomaga razumeti, kako se uporablja enega izmed L^AT_EXovih glavnih adutov. Na koncu poglavja se nahajajo tabele vseh matematičnih simbolov, ki so na voljo v L^AT_EXu.

Poglavje 4 obravnava izdelavo indeksov, urejanje seznama literature in vključevanje EPS slik. Predstavi tudi generiranje PDF dokumentov preko pdfL^AT_EXa in nekaj drugih priročnih razširitvenih paketov.

Poglavje 5 pokaže, kako lahko uporabimo L^AT_EX za kreiranje slik. Namesto tega, da sliko narišemo s kakšnim grafičnim programom, jo shranimo na datoteko in nato vključimo v L^AT_EX, jo lahko opišemo kar v L^AT_EXu, ki jo nariše za nas.

Poglavje 6 vsebuje nekaj potencialno nevarnih informacij o tem, kako lahko spremenimo standardno obliko dokumenta, narejenega z L^AT_EXom. Tu boste izvedeli, kako lahko stvari nastavimo tako, da se čudovita oblika L^AT_EXa spremeni v zanikrno ali bleščečo, odvisno od vaših sposobnosti.

Zelo pomembno je, da poglavja preberete v navedenem vrstnem redu. Knjiga konec koncev sploh ni tako dolga. Poskrbite za to, da pazljivo preberete zglede, saj se pomemben del podatkov skriva v različnih primerih, ki so razporejeni po celi knjigi.

L^AT_EX je na voljo za večino računalnikov, od PC in Mac do velikih UNIX in VMS sistemov. V teh navodilih se ne bomo ukvarjali s podrobnostmi namestitve in poganjanja L^AT_EXa na različnih platformah, temveč s tem, kako je potrebno napisati datoteko, ki jo želimo obdelati z L^AT_EXom.

Če potrebujete kakršno koli stvar povezano z L^AT_EXom, pogledjte na enega izmed obsežnih T_EXovih spletnih arhivov (Comprehensive T_EX Archive Network) oz. krajše (CTAN). Domača stran je <http://www.ctan.org>. Vse pakete lahko dobite tudi z ftp arhiva <ftp://www.ctan.org> oziroma z ene izmed njegovih številnih zrcalnih kopij po vsem svetu.

V knjigi boste našli še druge reference na CTAN, še posebno spletne naslove programov in dokumentov, ki jih lahko po potrebi prenesete na svoj računalnik. Namesto celotnih naslovov so le ti skrajšani v CTAN:, ki mu sledi ustrezna lokacija v CTAN drevesu.

Če nameravate poganjati L^AT_EX na vašem lastnem računalniku, si pogledjte, kaj je na voljo na CTAN:/tex-archive/systems.

Če imate kakršnokoli idejo o tem, kaj bi se dalo v tem dokumentu dodati, odstraniti oziroma popraviti, sporočite to za angleško različico na naslov

Tobias Oetiker <oetiker@ee.ethz.ch>

Department of Electrical Engineering,
Swiss Federal Institute of Technology

oziroma na

Bor Plestenjak <bor.plestenjak@fmf.uni-lj.si>

Oddelek za matematiko, Fakulteta za matematiko in fiziko,
Jadranska 19, 1000 Ljubljana

za slovensko različico. Še posebno so dobrodošle povratne informacije začetnikov v L^AT_EXu o tem, kateri deli so dobro razumljivi in kaj bi se dalo še bolje razložiti.

Zadnja različica angleškega dokumenta je na voljo na
[CTAN:/tex-archive/info/lshort](http://www.ctan.org/tex-archive/info/lshort)

Kazalo

Zahvala	iii
Predgovor	v
1 Kaj vse je dobro vedeti	1
1.1 Imena programov	1
1.1.1 T _E X	1
1.1.2 L ^A T _E X	1
1.2 Osnove	2
1.2.1 Avtor, knjižni oblikovalec in črkostavec	2
1.2.2 Oblikovanje dokumenta	2
1.2.3 Prednosti in slabosti	3
1.3 L ^A T _E Xova vhodna datoteka	4
1.3.1 Presledki	4
1.3.2 Posebni znaki	4
1.3.3 Ukazi v L ^A T _E Xu	5
1.3.4 Komentarji	6
1.4 Struktura vhodne datoteke	6
1.5 Tipično poganjanje v ukazni vrstici	7
1.6 Oblika dokumenta	9
1.6.1 Vrste dokumentov	9
1.6.2 Paketi	11
1.6.3 Oblika strani	11
1.7 Datoteke, ki jih lahko srečamo	11
1.8 Obsežnejši projekti	14
2 Stavljenje teksta	17
2.1 Zgradba teksta in jezika	17
2.2 Prelomi vrstic in strani	19
2.2.1 Obojestransko poravnani odstavki	19
2.2.2 Deljenje besed	20
2.3 Že pripravljene nize	21
2.4 Specialni znaki in simboli	21

2.4.1	Narekovaji	21
2.4.2	Pomišljaji in vezaji	22
2.4.3	Tilda (\sim)	22
2.4.4	Znak za stopinje (\circ)	22
2.4.5	Znak za valuto evro (€)	22
2.4.6	Tri pike (\dots)	23
2.4.7	Ligature	24
2.4.8	Akcenti in posebni znaki	24
2.5	Podpora drugih jezikov	24
2.5.1	Podpora za slovenščino	27
2.6	Razmiki med besedami	28
2.7	Naslovi, poglavja in razdelki	29
2.8	Sklicevanja	31
2.9	Opombe	31
2.10	Poudarjene besede	32
2.11	Okolja	32
2.11.1	Seznami	33
2.11.2	Levo, desno in sredinsko poravnan tekst	33
2.11.3	Navedki in verzi	34
2.11.4	Povzetek	35
2.11.5	Dobesedni izpis	35
2.11.6	Tabele	36
2.12	Plavajoči objekti	38
2.13	Zaščita krhkih ukazov	40
3	Stavljenje matematičnih formul	43
3.1	Uvod	43
3.2	Združevanje v matematičnem načinu	45
3.3	Osnovni gradniki matematičnih formul	46
3.4	Presledki v matematičnem načinu	50
3.5	Navpično poravnavanje	50
3.6	Fantomski objekti	52
3.7	Velikost pisave v matematičnem načinu	53
3.8	Izreki, trditve, \dots	54
3.9	Krepki simboli	56
3.10	Seznam matematičnih simbolov	57
4	Posebnosti	65
4.1	Vključevanje EPS slik	65
4.2	Literatura	67
4.3	Stvarno kazalo	68
4.4	Prilagojene glave strani	69
4.5	Paket verbatim	71
4.6	Prenos in namestitve dodatnih \LaTeX paketov	71

4.7	Uporaba orodja pdfL ^A T _E X	72
4.7.1	PDF dokumenti za svetovni splet	73
4.7.2	Pisave	73
4.7.3	Uporaba zunanjih slik	75
4.7.4	Hipertekstovne povezave	76
4.7.5	Težave s povezavami	78
4.7.6	Težave z zaznamki	79
4.8	Priprava elektronskih predstavitev	81
5	Priprava matematičnih slik	85
5.1	Kratki pregled	85
5.2	Okolje <code>picture</code>	86
5.2.1	Osnovni ukazi	86
5.2.2	Daljice	87
5.2.3	Puščice	88
5.2.4	Krožnice	89
5.2.5	Besedila in formule	90
5.2.6	<code>\multiput</code> in <code>\linethickness</code>	90
5.2.7	Ovali	91
5.2.8	Debelina črt	91
5.2.9	Večkratna uporaba vnaprej pripravljenih škatel s slikami	92
5.2.10	Kvadratne Bézierove krivulje	93
5.2.11	Verižnica	94
5.2.12	Hitrost v posebni teoriji relativnosti	95
5.3	<code>Xy-pic</code>	95
6	Prilagajanje L^AT_EXa	99
6.1	Novi ukazi, okolja in paketi	99
6.1.1	Novi ukazi	100
6.1.2	Nova okolja	101
6.1.3	Presledki na začetku in po koncu okolja	101
6.1.4	Poganjanje L ^A T _E Xa iz ukazne vrstice	102
6.1.5	Lastni paketi	103
6.2	Pisave in velikosti črk	103
6.2.1	Ukazi za spreminjanje pisave	103
6.2.2	Nevarnost na vidiku	106
6.2.3	Nasvet	107
6.3	Presledki	107
6.3.1	Razmik med vrsticami	107
6.3.2	Oblikovanje odstavka	107
6.3.3	Vodoravni razmiki in zapolnjevalci	108
6.3.4	Navpični presledki	110
6.4	Oblika strani	110
6.5	Še več zabave z dolžinami	112

6.6 Škatle	113
6.7 Poljubne črte	115
Literatura	117
Stvarno kazalo	119

Slike

1.1	Minimalna \LaTeX datoteka.	7
1.2	Realističen primer članka za revijo.	7
2.1	Primer vhodne \LaTeX datoteke v slovenščini.	28
4.1	Zgled uporabe paketa <code>fancyhdr</code>	70
4.2	Preprost zgled za razred <code>beamer</code>	82
6.1	Zgled paketa.	103
6.2	Parametri oblike strani.	111

Tabele

1.1	Vrste dokumentov.	9
1.2	Opcije za vrste dokumentov.	10
1.3	Nekaj paketov, ki se namestijo poleg \LaTeX a.	12
1.4	Vnaprej pripravljene oblike strani v \LaTeX u.	13
2.1	Polna vreča znakov za evro	23
2.2	Akcenti in posebni znaki.	25
2.3	Posebni znaki v slovenščini.	28
2.4	Oznake za dovoljene položaje plavajočega objekta.	38
3.1	Matematični akcenti.	57
3.2	Grške črke.	57
3.3	Binarne relacije.	58
3.4	Binarne operacije	58
3.5	Veliki operatorji.	59
3.6	Puščice.	59
3.7	Oklepaji.	59
3.8	Veliki oklepaji.	60
3.9	Razni simboli.	60
3.10	Nematematični simboli.	60
3.11	AMS oklepaji.	60
3.12	AMS grške in hebrejske črke.	60
3.13	AMS binarne relacije.	61
3.14	AMS puščice.	62
3.15	AMS negirane binarne relacije in puščice.	63
3.16	AMS binarne operacije.	63
3.17	AMS razni simboli.	64
3.18	Matematična abeceda.	64
4.1	Ključne opcije za paket <code>graphicx</code>	66
4.2	Zgledi vnosa gesla za stvarno kazalo.	68
6.1	Pisave.	104
6.2	Velikosti črk.	104

6.3	Absolutna velikost pisave v standardnih razredih.	105
6.4	Matematične pisave.	105
6.5	\TeX Merske enote.	109

Poglavje 1

Kaj vse je dobro vedeti

Na začetku tega poglavja je kratek pregled filozofije in zgodovine $\LaTeX 2_{\epsilon}$. V preostanku poglavja je poudarek na osnovni zgradbi dokumentov v \LaTeX u. Ko boste prebrali poglavje, boste v grobem vedeli, kako \LaTeX deluje, kar boste potrebovali, če boste želeli razumeti preostanek te knjige.

1.1 Imena programov

1.1.1 \TeX

\TeX je računalniški program, ki ga je razvil Donald E. Knuth [2]. Namenjen je za oblikovanje in izpis tekstov in matematičnih formul. Knuth je začel pisati tolmač za \TeX leta 1977, da bi izkoristil možnost digitalnega tiskanja, ki je v tem obdobju prodiralo med založniško industrijo. Še posebno je upal, da bo lahko zaustavil trend nazadovanja tiskarske kvalitete, ki ga je opazil pri svojih knjigah in člankih. \TeX v današnji obliki je nastal leta 1982, leta 1989 pa so drobni popravki omogočili boljšo podporo 8-bitnim znakom in različnim (naravnim) jezikom. \TeX je znan po tem, da je izredno stabilen, da teče na različnih operacijskih sistemih in da je skoraj brez napak. Oznaka različice \TeX a konvergira k π in je trenutno 3.141592.

\TeX se izgovarja »teh«, kot na primer v besedi »tehnologija«. V ASCII okolju \TeX postane **TeX**.

1.1.2 \LaTeX

\LaTeX je paket dodatnih ukazov za \TeX , ki omogočajo avtorjem oblikovanje in izpisovanje njihovih dokumentov v najvišji tiskarski kvaliteti s pomočjo že pripravljenih profesionalnih vzorcev. V \LaTeX u lahko še naprej uporabljamo skoraj vse \TeX ove ukaze. \LaTeX je prvotno napisal Leslie Lamport [1], sedaj pa zanj skrbi Frank Mittelbach.

\LaTeX se izgovori kot »La-teh«. V ASCII okolju \LaTeX postane LaTeX. $\LaTeX 2_{\epsilon}$ se izgovori kot »La-teh dva e« in se v ASCII okolju piše kot LaTeX2e.

1.2 Osnove

1.2.1 Avtor, knjižni oblikovalec in črkostavec

Ko želi kaj objaviti, avtor pošlje svoj rokopis založniku. Tam knjižni oblikovalec določi obliko dokumenta (širino stolpcev, obliko črk, velikost presledkov pred in po naslovih, ...). Oblikovalec svoja navodila skupaj z rokopisom preda črkostavcu, ki na podlagi teh podatkov stavi knjigo.

Človeški knjižni oblikovalec poskuša dojeti, kaj je avtor mislil, ko je pisal rokopis. Glede na njegove poklicne izkušnje in glede na vsebino rokopisa se odloči za naslove poglavij, citate, zglede, formule, itd.

V \LaTeX okolju \LaTeX prevzame vlogo knjižnega oblikovalca, za stavljenje pa uporablja \TeX . Ker pa je \LaTeX »samo« računalniški program, potrebuje več vodenja. Avtor mora poskrbeti za dodatne podatke, ki opisujejo logično strukturo njegovega dela. Ti podatki se vpišejo med tekst v obliki \LaTeX ukazov.

To je povsem različen način dela od pristopa WYSIWYG¹, ki ga uporablja večina sodobnih urejevalnikov besedil, kot sta npr. *MS Word* in *Corel WordPerfect*. V teh programih avtor določi obliko dokumenta interaktivno med samim vnosom besedila v računalnik. Med celotnim procesom na zaslonu vidi, kakšna bo končna oblika, ko bo dokument natisnjen.

Pri \LaTeX u ni možno videti končne oblike med samim tipkanjem besedila. Lahko pa vidimo končno obliko na zaslonu, potem ko datoteko prevedemo z \LaTeX om. Tako lahko popravke naredimo še preden dokument izpišemo s tiskalnikom.

1.2.2 Oblikovanje dokumenta

Oblikovanje besedila je prava spretnost. Neizkušeni avtorji pogosto naredijo resno napako, ko privzamejo, da je oblikovanje knjige vprašanje estetike (če ima dokument estetski zunanji videz, potem naj bi bil dobro oblikovan). Dokument je namenjen branju in ne temu, da visi v umetniški galeriji, zato sta čitljivost in razumljivost veliko pomembnejša od čudovitega izgleda. Npr.:

- Velikost črk in oštevilčenje naslovov je potrebno izbrati tako, da je struktura poglavij in razdelkov jasna razvidna.
- Vrstice morajo biti po eni strani dovolj kratke, da ne utrujajo oči bralcev, po drugi strani pa dovolj dolge, da lepo napolnijo strani.

1. What you see is what you get oz. Kar vidiš, to tudi dobiš.

Z WYSIWYG sistemi avtorji pogosto generirajo vsečne dokumente, ki pa so ali zelo slabo strukturirani ali pa imajo nepovezano strukturo. \LaTeX preprečuje tovrstne oblikovalske napake s tem, ko avtorja sili, da določi *logično* strukturo njegovega dokumenta, \LaTeX pa potem izbere najprimernejšo obliko.

1.2.3 Prednosti in slabosti

Ko se srečajo uporabniki WYSIWYG programov z uporabniki \LaTeX a, ponavadi razglabljaajo o »prednostih \LaTeX a pred normalnimi urejevalniki besedil« in obratno. Pri tovrstnih diskusijah je najbolje ostati tiho, saj se pogosto izrodijo. Toda vedno se jim le ne morete izogniti . . .

. . .in zato je tu nekaj iztočnic. Glavne prednosti \LaTeX a pred normalnimi urejevalniki besedil so:

- Na voljo so profesionalno oblikovani vzorci, zaradi katerih je končni rezultat tak, kot da bi res prišel iz tiskarne.
- Stavljenje matematičnih formul je podprto na preprost način.
- Uporabnik se mora naučiti le nekaj preprosto razumljivih ukazov, ki določajo logično strukturo dokumenta. Uporabniku skoraj nikoli ni potrebno brkljati po aktualni obliki dokumenta.
- Zapletene strukture, kot so npr. opombe pod črto, reference, kazala in sezname literature, se dajo narediti na preprost način.
- Obstajajo številni prosto dosegljivi dodatni paketi za različne tiskarske naloge, ki niso direktno podprte z osnovnim \LaTeX om. Tako npr. obstajajo paketi za vključevanje POSTSCRIPT slik in paketi za stavljenje seznama literature po določenih standardih. Številni dodatni paketi so opisani v *The \LaTeX Companion* [3].
- \LaTeX spodbuja avtorje, da pišejo dobro strukturirane tekste, saj je specifikacija strukture eden izmed osnovnih principov delovanja \LaTeX a.
- \TeX , na katerem sloni $\LaTeX 2_{\epsilon}$, je zelo prenosljiv in prosto dostopen, kar pomeni, da sistem lahko poganjamo v skoraj vseh računalniških okoljih.

\LaTeX ima seveda tudi nekaj slabosti, in čeprav sam težko poiščem kakšno smiselno, sem prepričan, da jih drugi poznajo na stotine ; -)

- \LaTeX ni primeren za ljudi, ki so prodali svojo dušo . . .

- Čeprav se da nekaj parametrov nastaviti znotraj privzetih oblik dokumentov, je načrtovanje popolnoma novega vzorca zapleteno opravilo, za katerega porabimo veliko časa.²
- Težko je pisati nestrukturirane in neorganizirane dokumente.
- Spodbudnim začetnim korakom navkljub nikoli ne boste v celoti dojeli koncepta logičnih oznak.

1.3 L^AT_EXova vhodna datoteka

Vhodni podatek za L^AT_EX je navadna ASCII tekstovna datoteka. Datoteko lahko sestavimo v poljubnem ASCII urejevalniku. Datoteka poleg besedila dokumenta vsebuje še ukaze, ki povedo L^AT_EXu, kako naj oblikuje tekst.

1.3.1 Presledki

Vse »prazne« znake, kot sta presledek in tabulator, L^AT_EX obravnava enako kot presledek. *Zaporedni* presledki se obravnavajo kot *en* presledek. Vodilni presledki na začetku vrstice se na splošno ignorirajo, konec vrstice pa se obravnava kot en presledek.

Prazna vrstica med dvema vrsticama teksta pomeni konec odstavka, pri čemer se *zaporedne* prazne vrstice obravnava kot *eno* prazno vrstico. Zgled je priložen spodaj. Na levi strani je tekst v vhodni datoteki, na desni strani pa prevedeni rezultat.

Ni pomembno, če za besedo
stoji en ali pa več presledkov.

Prazna vrstica začenja nov
odstavek.

Ni pomembno, če za besedo stoji en ali pa
več presledkov.

Prazna vrstica začenja nov odstavek.

1.3.2 Posebni znaki

Naslednji simboli so rezervirani znaki, ki ali imajo poseben pomen v L^AT_EXu ali pa niso na voljo v vseh naborih znakov. Če naslednje znake direktno vstavimo v tekst, potem ne samo, da se ne bodo izpisali, v večini primerov bo L^AT_EX vrnil nepričakovane rezultate.

\$ % ^ & _ { } ~ \

Kot vidite v nadaljevanju, lahko tudi te znake uporabljamo v svojih dokumentih, če prednje postavimo znak \ (backslash):

² Govori se, da bo to ena izmed pglavitnih reči, ki bo rešena v nastajajočem L^AT_EX3 sistemu.

```
\# \$ \% \^{} \& \_ \{ \} \~{}
```

```
# $ % ^ & _ { } ~
```

Ostale simbole in še več lahko natisnemo s posebnimi ukazi v matematičnih formulah ali pa kot akcente. Znak `\ ne` moremo vnesti kot `(\)`, saj je to zaporedje rezervirano za zlom vrstice.³

1.3.3 Ukazi v \LaTeX u

Ukazi v \LaTeX u so občutljivi na velike in male črke in imajo eno izmed naslednjih dveh oblik:

- Začnejo se z znakom `\`, temu pa sledi ime sestavljeno le iz črk. Imena ukazov se končajo ali s presledkom, številko ali kakšnim drugim znakom, ki ni črka.
- Sestavljeni so iz znaka `\` in natančno enega posebnega znaka (lahko je tudi številka).

\LaTeX ignorira presledke za ukazi. Če želite presledek za ukazom, potem morate na konec ukaza dati ali `{}` in presledek ali pa posebni ukaz za presledek. `{}` prepreči \LaTeX u, da bi pojedel vse presledke za imenom ukaza.

```
Bral sem, da Knuth deli ljudi,
ki uporabljajo \TeX{} na
\TeX{}nike in \TeX{} perte.\.
Danes je \today.
```

```
Bral sem, da Knuth deli ljudi, ki uporabljajo
TeX na TeXnike in TeXperte.
Danes je 8. maj 2008.
```

Nekateri ukazi potrebujejo še parameter, ki ga lahko podamo med zavitimi oklepaji `{ }` za imenom ukaza. Nekateri ukazi podpirajo tudi opsijske parametre, ki jih dodamo za imenom ukaza med oglatimi oklepaji `[]`. V naslednjem zgledu so uporabljeni nekateri \LaTeX ovi ukazi. Nič hudega, če jih ne razumete, saj bodo razloženi v nadaljevanju.

```
Lahko se \textsl{zanesete} name!
```

```
Lahko se zanesete name!
```

```
Začni z novo vrstico
točno tukaj!\newline
Hvala!
```

```
Začni z novo vrstico točno tukaj!
Hvala!
```

3. Raje poskusite `\backslash`, ki res vrne `\`.

1.3.4 Komentarji

Kadar \LaTeX v vhodni datoteki sreča znak %, ignorira vse znake v nadaljevanju tekoče vrstice, zlom vrstice in vse presledke na začetku naslednje vrstice.

To lahko uporabljamo za vnos komentarjev v vhodno datoteko, ki jih ne bo na končnem izpisu.

```
To je % neumen
% bolje: poučen <----
primer: Otorino%
        laringo%
log
```

To je primer: Otorinolaringolog

Z znakom % lahko zlomimo dolgo vhodno vrstico v primeru, ko ne moremo uporabiti presledka ali prehoda v novo vrstico.

Za daljše komentarje je bolje uporabiti okolje `comment`, ki je definirano v paketu `verbatim`. To pomeni, da moramo, če želimo uporabljati okolje `comment`, v preambulo dokumenta dodati ukaz `\usepackage{verbatim}`.

```
To je nov
\begin{comment}
precej neumen,
a koristen
\end{comment}
primer vstavljanja
komentarja v dokument.
```

To je nov primer vstavljanja komentarja v dokument.

Upoštevati je potrebno, da naštetu ne deluje znotraj kompleksnih okolij, kot je npr. okolje za zapis matematičnih formul.

1.4 Struktura vhodne datoteke

Ko $\text{\LaTeX} 2_{\epsilon}$ prevaja vhodno datoteko, pričakuje, da ima ta določeno strukturo. Tako se mora vsaka vhodna datoteka začeti z ukazom

```
\documentclass{...}
```

Tu določimo vrsto dokumenta, ki ga pišemo. Sledijo lahko ukazi, ki vplivajo na stil celotnega dokumenta, lahko pa tudi vključimo dodatne pakete, ki \LaTeX razširijo z novimi možnostmi. Paket vključimo z ukazom

```
\usepackage{...}
```

Ko končamo z nastavitvami,⁴ začnemo telo dokumenta z ukazom

4. Del datoteke med `\documentclass` in `\begin{document}` se imenuje *preambula*.

```
\begin{document}
```

Tu sedaj vnašamo tekst pomešan z različnimi koristnimi L^AT_EX ukazi. Na koncu dokumenta mora stati ukaz

```
\end{document}
```

ki pove L^AT_EXu, da je konec dela. Če temu ukazu sledi še kaj, bo L^AT_EX to ignoriral.

Slika 1.1 kaže vsebino minimalne L^AT_EX 2_ε datoteke. Za malenkost zapletenejša vhodna datoteka je predstavljena na sliki 1.2. Vsi ukazi iz tega zglada bodo razloženi v nadaljevanju.

1.5 Tipično poganjanje v ukazni vrstici

Gotovo vas že zelo mika preizkusiti čisto kratek primer L^AT_EX vhodne datoteke s strani 7. Tukaj je kratka pomoč: L^AT_EX sam ne vsebuje nobenega grafičnega vmesnika in nobenih razkošnih gumbov, na katere bi pritiskali.

```
\documentclass{article}
\begin{document}
Majhno je lepo.
\end{document}
```

Slika 1.1: Minimalna L^AT_EX datoteka.

```
\documentclass[a4paper,11pt]{article}
% definiramo naslov
\author{H.~Partl}
\title{Minimalizem}
\begin{document}
% vstavimo naslov
\maketitle
% vstavimo kazalo
\tableofcontents
\section{Začetek}
Tukaj se začne moj dragi tekst.
\section{Konec}
\ldots{} in tukaj ga je konec.
\end{document}
```

Slika 1.2: Realističen primer članka za revijo.

To je le program, ki prežveči vašo vhodno datoteko. Nekatere namestitve \LaTeX a vsebujejo tudi grafični vmesnik na katerem lahko kliknemo in tako prevedemo našo vhodno datoteko. Tukaj bomo pogledali, kako pripravimo \LaTeX do prevajanja vhodne datoteke preko ukazne vrstice. Pri tem bomo predpostavili, da je na našem računalniku že nameščen delujoči \LaTeX .

1. Najprej uredimo oz. ustvarimo \LaTeX vhodno datoteko. To mora biti navadna ASCII datoteka. V Oknih je dobro preveriti, če bo datoteka res shranjena v ASCII oziroma *Plain Text* obliki (primerna sta npr. programa *Notepad* in *Textpad*), na Unixu pa drugega niti ne moremo dobiti. Ko izbiramo ime datoteke, se prepričajmo, da ima končnico `.tex`, npr. `foo.tex`.
2. Poženemo \LaTeX na vhodni datoteki. V primeru uspeha bomo dobili `.dvi` datoteko `foo.dvi`, sicer pa se bo \LaTeX ustavil in javil napako. Datoteko je ponavadi potrebno prevesti večkrat, da se uredijo notranje reference, kazalo in citiranje literature.

```
latex foo.tex
```

3. Sedaj lahko na zaslonu pogledamo DVI datoteko z ustreznim programom

```
yap foo.dvi
```

Če delamo na Unix sistemu, potem za pregled `.dvi` datotek uporabljamo program `xdvi`, v Oknih pa poleg `yap` lahko uporabljamo še `dviwin`.

DVI datoteko lahko prevedemo v PostScript, ki ga lahko natisnemo ali gledamo s programom Ghostscript.

```
dvips -Pcmz foo.dvi -o foo.ps
```

Če imamo srečo, je v naši namestitvi \LaTeX sistema tudi orodje `dvipdf`, s katerim lahko DVI datoteko direktno spremenimo v pdf datoteko.

```
dvipdf foo.dvi
```

1.6 Oblika dokumenta

1.6.1 Vrste dokumentov

Prva informacija, ki jo \LaTeX potrebuje za prevajanje vhodne datoteke, je vrsta dokumenta, ki jo želi avtor narediti. To je določeno z ukazom `\documentclass`.

```
\documentclass[opcije]{razred}
```

Tu *razred* določa vrsto dokumenta. Tabela 1.1 prikazuje vrste dokumentov, ki so opisane v teh navodilih. Distribucija $\LaTeX 2_\epsilon$ vsebuje še dodatne razrede za druge vrste dokumentov, kot so npr. pisma in prosojnice. S parametri *opcije* lahko prilagodimo obnašanje vrste dokumenta. Če uporabimo več kot eno opcijo, jih moramo ločiti z vejicami. Najpogostejše opcije za standardne vrste dokumentov so predstavljene v tabeli 1.2.

Zgled: Vhodna \LaTeX datoteka se lahko začne z

```
\documentclass[11pt,twoside,a4paper]{article}
```

in to je navodilo \LaTeX u, da naj stavi dokument kot *članek* z osnovno velikostjo črke *11 pik*, prevedeni dokument pa naj bo primeren za *dvostransko tiskanje na papirju velikosti A4*.

Tabela 1.1: Vrste dokumentov.

article	za članke v znanstvenih revijah, predstavitev, kratka poročila, dokumentacije programov, povabila, ... ,
proc	za poročila referatov, zasnovan je na razredu article,
minimal	je najbolj osnoven razred. Nastavi le velikost strani in osnovni nabor črk. Uporablja se v glavnem pri odkrivanju in odpravljanju napak v programu.
report	za daljša poročila z več poglavji, manjše knjige, disertacije, ...
book	za prave knjige,
letter	za pisma,
slides	za prosojnice. Ta razred uporablja večje znake v pisavi sans serif. Namesto tega lahko uporabljate paket <code>FoilTeX</code> ^a .

a. CTAN:/tex-archive/macros/latex/contrib/supported/foiltex

Tabela 1.2: Opcije za vrste dokumentov.

<code>10pt, 11pt, 12pt</code>	Nastavi osnovno velikost črk v dokumentu. Če te opcije ne navedemo, je privzeta vrednost <code>10pt</code> .
<code>a4paper, letterpaper, ...</code>	Nastavi velikost papirja. Privzeta vrednost je <code>letterpaper</code> , poleg tega pa imamo še <code>a5paper</code> , <code>b5paper</code> , <code>executivepaper</code> in <code>legalpaper</code> . Ker pri nas uporabljamo papir A4, je to potrebno vedno nastaviti.
<code>fleqn</code>	S to nastavitvijo se formule stavijo levo namesto sredinsko poravnane.
<code>leqno</code>	Številčenje enačb je na levi strani namesto na desni.
<code>titlepage, notitlepage</code>	Določa, ali naj naslovu dokumenta sledi nova stran ali ne. Razred <code>article</code> se privzeto ne začne z novo stranjo, razreda <code>report</code> in <code>book</code> pa se.
<code>onecolumn, twocolumn</code>	Določa, ali naj $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ stavi dokument z enim ali dvema stolpcema.
<code>twoside, oneside</code>	Določa, ali bo končni izpis natisnjen enostransko ali dvostransko. Razreda <code>article</code> in <code>report</code> sta privzeto enostranska, razred <code>book</code> pa je dvostranski. Ta opcija vpliva le na stil dokumenta, <i>ne</i> vpliva pa na to, ali bo tiskalnik izpisal tekst eno ali dvostransko.
<code>landscape</code>	Spremeni obliko strani iz pokončne v ležečo obliko.
<code>openright, openany</code>	Ali naj se poglavja začenjajo vedno na desni strani ali na prvi naslednji prazni strani. To ne deluje v razredu <code>article</code> , saj ta razred nima poglavij. Razred <code>report</code> privzeto začinja poglavja na naslednji strani, razred <code>book</code> pa na naslednji desni strani.
<code>final, draft</code>	Ali gre za končno ali delovno verzijo. Pri delovni verziji so vse predolge vrstice označene s črnimi pravokotniki.

1.6.2 Paketi

Med pisanjem svojih dokumentov boste verjetno odkrili, da nekaterih stvari ne morete narediti z osnovnim L^AT_EXom. Če želite vključiti slike, barvni tekst, ali vključiti izvorno kodo iz neke datoteke v svoj tekst, morate razširiti zmožnosti L^AT_EXa. Tovrstne izboljšave so paketi, vključimo pa jih z ukazom

```
\usepackage[opcije]{paket}
```

kjer je *paket* ime paketa, *opcije* pa seznam ključnih besed, ki nastavljajo določene opcije v paketu. Veliko paketov dobimo že ob osnovni namestitvi L^AT_EX 2_ε (poglej tabelo 1.3). Ostali paketi so na voljo ločeno. Več informacij o nameščenih paketih na vašem računalniku lahko dobite v vašem *Local Guide* [5]. Glavni vir podatkov o L^AT_EX paketih je *The L^AT_EX Companion* [3]. Vsebuje opise na stotine paketov skupaj z navodili, kako lahko razširimo L^AT_EX 2_ε z novimi ukazi.

1.6.3 Oblika strani

Posamezna stran je sestavljena iz glave, telesa in dna. L^AT_EX podpira tri vnaprej pripravljene kombinacije za glavo in dno, t.i. oblike strani. Parameter *stil* ukaza

```
\pagestyle{stil}
```

določa, kateri naj se uporablja. Tabela 1.4 vsebuje seznam že pripravljenih oblik strani.

Možno je spremeniti obliko strani le za tekočo stran z ukazom

```
\thispagestyle{stil}
```

Navodila za to, kako lahko naredimo lastno glavo in dno lahko najdete v *The L^AT_EX Companion* [3] in v razdelku 4.4 na strani 69.

1.7 Datoteke, ki jih lahko srečamo

Ko delamo z L^AT_EXom se hitro znajdemo v zmešnjavi datotek z različnimi končnicami, nimamo pa nobenega pojma, čemu služijo. Sledi seznam z različnimi vrstami datotek, ki jih lahko srečamo med delom s T_EXom. Seznam ni popoln in lahko srečamo še kakšne druge datoteke.

.tex L^AT_EX ali T_EX vhodna datoteka. Prevedemo jo z **latex**.

Tabela 1.3: Nekaj paketov, ki se namestijo poleg L^AT_EXa.

<code>doc</code>	Omogoča dokumentacijo L ^A T _E X programov. Opisan je v <code>doc.dtx</code> ^a in v <i>The L^AT_EX Companion</i> [3].
<code>exscale</code>	Poskrbi za to, da je velikost matematičnih simbolov usklajena z velikostjo znakov. Opisan je v <code>ltxscale.dtx</code> .
<code>fontenc</code>	Določa, katero kodno tabelo naj L ^A T _E X uporablja. Opisan je v <code>loutenc.dtx</code> .
<code>ifthen</code>	Omogoča pogojne stavke oblike ‘if... then do... otherwise do...?’ Opisan je v <code>ifthen.dtx</code> in v <i>The L^AT_EX Companion</i> [3].
<code>latexsym</code>	S tem paketom pridemo do dodatnih simbolov iz L ^A T _E Xovega nabora simbolov. Opisan je v <code>latexsym.dtx</code> in v <i>The L^AT_EX Companion</i> [3].
<code>makeidx</code>	Doda ukaze za izdelavo indeksov. Opisan je v razdelku 4.3 in v <i>The L^AT_EX Companion</i> [3].
<code>syntonly</code>	Prevede dokument brez stavljenja.
<code>inputenc</code>	S tem paketom lahko določimo vhodno kodno tabelo kot so npr. ASCII, ISO Latin-1, ISO Latin-2, 437/850 IBM kodni tabeli, Apple Macintosh, Next, ANSI-Windows ali kaj poljubnega. Paket je opisan v <code>inputenc.dtx</code> .

^a. Ta datoteka naj bi bila nameščena na vašem sistemu, vi pa iz nje lahko dobite `dvi` datoteko če odtipkate `latex doc.dtx` v enem izmed imenikov, kjer imate dovoljenje za pisanje. Isto velja za vse ostale datoteke `*.dtx` v tej tabeli.

- .sty** \LaTeX paket makro ukazov. To datoteko vključimo v \LaTeX dokument z ukazom `\usepackage`.
- .dtx** Dokumentirani \TeX . To je glavna oblika širjenja \LaTeX ovih paketov. Če prevedemo `.dtx` datoteko, dobimo dokumentirane makro ukaze v \LaTeX paketu vsebovanem v `.dtx` datoteki.
- .ins** Namestitev za datoteke v ustrezni `.dtx` datoteki. Če z interneta naložimo \LaTeX paket, potem ponavadi dobimo `.dtx` in `.ins` datoteko. Če poženemo \LaTeX na `.ins` datoteki, potem iz `.dtx` datoteke dobimo potrebne datoteke.
- .cls** Razredne datoteke določajo obliko dokumentov. Te datoteke izberemo z ukazom `\documentclass`.
- .fd** Opisi novih pisav.

Naslednje datoteke dobimo, ko poženemo \LaTeX na vhodni datoteki:

- .dvi** (Device Independent file). To je glavni rezultat prevajanja z \LaTeX om. Vsebino lahko pogledamo s programom za prikazovanje DVI datotek ali pa pošljemo v izpis na tiskalnik s pomočjo `dvips` ali podobnega programa.
- .log** Vrne podroben opis dogajanja med zadnjim prevajanjem (tu je tudi seznam napak v primeru le teh).
- .toc** Tu so shranjeni vsi naslovi poglavij. Prebere se pri vsakem zagonu prevajalnika in se porabi za generiranje kazala.
- .lof** Podobno kot `.toc` le da gre za kazalo slik.
- .lot** Podobno kot prej za kazalo tabel.

Tabela 1.4: Vnaprej pripravljene oblike strani v \LaTeX u.

plain izpisuje številke strani na dnu strani na sredini. To je privzeta oblika strani.

headings izpisuje trenutni naslov poglavja in številko strani v glavi vsake strani, dno pa je prazno. (Ta oblika je uporabljena v tem dokumentu.)

empty nastavi, da sta tako glava kot dno strani prazna.

- .aux** Datoteka, kjer se shranjujejo podatki od enega prevajanja do drugega. Med drugim se tu shranijo tudi vse notranje reference.
- .idx** Če dokument vsebuje indeks, potem \LaTeX sem shrani vse besede, ki gredo v indeks. To datoteko je potrebno prevesti s programom `makeindex`. Poglejte razdelek 4.3 na strani 68 za več informacij o indeksih.
- .ind** To je rezultat programa `makeindex` in to je datoteka z indeksom pripravljena za to, da se vstavi pri naslednjem prevajanju datoteke.
- .ilg** Opis dogajanja pri zadnjem prevajanju z `makeindex`.

1.8 Obsežnejši projekti

Kadar delamo z obsežnimi dokumenti, lahko razdelimo vhodno datoteko na več delov. \LaTeX ima za to namenjena dva ukaza.

```
\include{filename}
```

uporabimo v telesu dokumenta, če želimo vstaviti vsebino datoteke z imenom *filename.tex*. Pomembno je vedeti, da bo \LaTeX pred prevajanjem z datoteke *filename.tex* začel z novo stranjo.

Drugi ukaz lahko uporabljamo v preambuli. Z njim lahko \LaTeX u naročimo, da vključi vhodne datoteke le z nekaterih `\include` datotek.

```
\includeonly{datoteka, datoteka, ...}
```

Ko se ta ukaz izvede v preambuli, se izvedejo ukazi `\include` le za tiste datoteke, ki so navedene kot argumenti v `\includeonly`. Med imeni datotek in vejicami ne sme biti nobenih presledkov.

Ukaz `\include` začenja staviti vključeni tekst na novi strani. To je koristno kadar uporabljamo `\includeonly`, saj se prelomi strani ne spremenijo, tudi če izpustimo kakšno datoteko. V nekaterih primerih to ni zaželeno. V tem primeru lahko uporabimo ukaz

```
\input{datoteka}
```

Ta preprosto vključi navedeno datoteko.

Če želimo da \LaTeX na hitro preveri naš dokument lahko uporabimo paket `syntonly`. Ta povzroči, da \LaTeX na hitro oplazi dokument, saj preverja le pravilnost sintakse in uporabo ukazov, ne generira pa nobenega rezultata (DVI). V tem načinu \LaTeX teče hitreje in tako prihranimo dragocen čas. Uporaba je zelo preprosta:

```
\usepackage{syntonly}  
\syntonly
```

Ko želimo, da pri prevajanju res nastanejo strani, le zakomentiramo drugo vrstico (dodamo znak za procent na začetek).

Poglavje 2

Stavljenje teksta

Če ste prebrali prejšnje poglavje, potem poznate osnovne sestavine $\text{\LaTeX} 2_{\epsilon}$ datotek. V tem poglavju bomo to dopolnili z dodatnimi podatki, ki jih potrebujete, če želite pisati »ta prave« dokumente.

2.1 Zgradba teksta in jezika

Avtor: Hanspeter Schmid <hanspi@schmid-werren.ch>

Glavni namen pisanja teksta je prenesti določene ideje, podatke, ali znanje na bralca. Le ta bo tekst razumel bolje, če bodo ideje dobro strukturirane, to strukturo pa bo videl in občutil veliko bolje, če se bo v tiskani obliki zrcalila logična in semantična zgradba vsebine.

\LaTeX se razlikuje od ostalih urejevalnikov besedil v tem, da mu moramo podati logično in semantično zgradbo teksta. Na podlagi tega lahko iz pravil, ki veljajo za izbrani razred dokumenta in iz ustreznih stilskih datotek, prevede besedilo v obliko za tiskanje.

Najpomembnejša enota besedila v \LaTeX u (in pri stavljenju teksta) je odstavek. Odstavek predstavlja eno zaključeno misel ali eno idejo. V razdelkih, ki sledijo, se bomo naučili, kako lahko prelomimo vrstico (npr. z \backslash) in kako pridemo do novega odstavka (npr. tako, da v vhodnem besedilu pustimo prazno vrstico). Torej, če začnemo novo misel, potem naj se začne v novem odstavku, sicer pa v vhodni datoteki uporabljamo le prelome vrstic. Če smo v dvomih ali naj naredimo nov odstavek ali ne, potem si predstavljamo celotno besedilo kot tekoči trak idej in razmišljanj. Če imamo v besedilu nov odstavek, a se misel iz prejšnjega nadaljuje, potem v resnici ne potrebujemo novega odstavka. Če pa se znotraj odstavka pojavi popolnoma nova misel, potem je potrebno začeti nov odstavek.

Večina popolnoma podcenjuje pomembnost dobro postavljenih prelomov odstavkov. Veliko jih sploh ne pozna pomena odstavka ali pa, še posebno v \LaTeX u, dodajajo prelome odstavkov, ne da bi za to sploh vedeli. Ta zadržna napaka je še posebno lahko zagrešiti kadar tekst vsebuje tudi enačbe.

Poglejte naslednji zgled in ugotovite, zakaj nekje pred in po enačbi uporabljamo prazno vrstico (prelom odstavka), drugje pa ne. (Če še ne poznate dovolj ukazov, da bi lahko brez težav razumeli te zglede, preberite to in še naslednje poglavje, potem pa se vrnite na ta razdelek.)

```
\ldots Preko Pitagorovega izreka
\begin{equation}
c^2 = a^2+b^2
\end{equation}
lahko izračunamo tretjo
stranico pravokotnega trikotnika
iz ostalih dveh.
```

...Preko Pitagorovega izreka

$$c^2 = a^2 + b^2 \quad (2.1)$$

lahko izračunamo tretjo stranico pravokotnega trikotnika iz ostalih dveh.

```
\ldots iz česar sledi Kirchoffov
zakon o toku:
\begin{equation}
\sum_{k=1}^n I_k = 0 \ ; \ .
\end{equation}
```

```
Izpeljemo lahko tudi Kirchoffov
zakon o napetosti \ldots
```

... iz česar sledi Kirchoffov zakon o toku:

$$\sum_{k=1}^n I_k = 0 . \quad (2.2)$$

Izpeljemo lahko tudi Kirchoffov zakon o napetosti ...

```
\ldots, ki ima številne prednosti.
```

```
\begin{equation}
I_D = I_F - I_R
\end{equation}
je bistvo drugačnega modela
tranzistorja. \ldots
```

..., ki ima številne prednosti.

$$I_D = I_F - I_R \quad (2.3)$$

je bistvo drugačnega modela tranzistorja.
...

Naslednja manjša enota je stavek. V angleških besedilih za piko, ki konča stavek, stoji daljši presledek kot za piko, ki označuje okrajšavo. \LaTeX sicer sam poskuša ugotoviti, kaj od tega želimo, če pa mu to ne uspe zadovoljivo, mu moramo povedati, kaj želimo. To je razloženo v nadaljevanju tega poglavja.

Strukturiranje teksta se nadaljuje celo na dele stavkov. Večina jezikov ima zelo zapletena pravila za postavljanje ločil, v veliko jezikov (vključno z nemščino in angleščino), pa se da zadeti skoraj vsako vejico, če se zavedamo, da predstavlja kratko zaustavitev toka jezika. Če niste prepričani, kam postaviti vejico, preberite stavek naglas in za vsako vejico naredite malo premora. Če to na kakšnem mestu ne zveni najbolje, tisto vejico odstranite. Prav tako, če morate na kakšnem mestu zajeti zrak (ali se ustaviti), potem na to mesto postavite vejico.

Odstavke je potrebno logično povezati v strukture na višjem nivoju v poglavja, razdelke, podrazdelke, in tako dalje. Kako deluje ukaz kot je npr. `\section{Struktura teksta in jezika}` je tako očitno, da je skoraj samoumevno kako se uporablja te višje strukture.

2.2 Prelomi vrstic in strani

2.2.1 Obojestransko poravnani odstavki

Ponavadi so knjige stavljene tako, da imajo vse vrstice enako dolžino. \LaTeX vstavi potrebne prelome vrstic in presledke med besedami tako, da optimizira vsebino celotnega odstavka. Če je potrebno, tudi deli besede, ki jih ni možno udobno postaviti v vrstico. Sama oblika odstavkov je odvisna od uporabljenega razreda. Normalno je prva vrstica zamaknjena, med dvema odstavkoma pa ni dodatnega navpičnega presledka. Več podrobnosti o tem lahko najdete v razdelku [6.3.2](#).

V posebnih primerih je potrebno ukazati \LaTeX u, da naj prelomi vrstico ali stran. Ukaz

```
\ ali \newline
```

izsili prelom vrstice, ostanemo pa v istem odstavku. Ukaz

```
\*
```

dodatno preprečuje prelom strani za izsiljenim prelomom vrstice. Ukaz

```
\newpage
```

začenja novo stran. Ukazi

```
\linebreak[n], \nolinebreak[n], \pagebreak[n] and \nopagebreak[n]
```

naredijo to, kar pravijo njihova imena. Z dodatnim parametrom n lahko vplivamo na njihovo obnašanje. Vrednost parametra je lahko med 0 in 4. Če nastavimo n na vrednost pod 4, potem pustimo \LaTeX u možnost, da ignorira naš ukaz, če bo rezultat zgledal zelo slabo. Teh »break« ukazov ne smemo zamenjevati z »new« ukazi. Celo kadar uporabimo »break« ukaz, bo \LaTeX še vedno poskušal poravnati desni rob oziroma skupno dolžino strani, kot je opisano v naslednjem razdelku. Če res želimo novo vrstico (new line), potem je potrebno uporabiti ustrezni ukaz. Uganite njegovo ime!

\LaTeX vedno poskuša narediti čim boljše prelome vrstic. Če vseeno ne uspe zlomiti vrstice tako, da bi to zadoščalo njegovim visokim standardom,

potem pusti, da ena vrstica štrli ven na desni strani odstavka. V tem primeru \LaTeX med prevajanjem vhodne datoteke izpiše opozorilo (»overfull hbox«). Najpogostejši razlog je, da \LaTeX ni uspel poiskati primernege mesta za delitev besede.¹ \LaTeX u lahko naročimo, da malo omili svoj kriterij z ukazom `\sloppy`. Ta prepreči predolge vrstice tako, da poveča presledke med posameznimi besedami, pa čeprav končni izgled ni optimalen. Tudi v tem primeru \LaTeX pri prevajanju izpiše opozorilo (»underfull hbox«). V večini takšnih primerov rezultat ne zgleda najboljše. Ukaz `\fussy` vrne \LaTeX nazaj na privzeto nastavitvev.

2.2.2 Deljenje besed

\LaTeX deli besede, kadar je to potrebno. Če algoritem za deljenje ne najde pravičnega mesta deljenja, lahko takšno situacijo popravimo tako, da \TeX u povemo za to izjemo.

Ukaz

```
\hyphenation{seznam besed}
```

povzroči, da se besede, ki so navedene v seznamu, delijo le na mestih označenih z »-«. Argument ukaza naj vsebuje samo besede zgrajene iz normalnih črk oziroma znakov, ki se v trenutnem kontekstu obnašajo kot normalne črke. Namigi za deljenje se shranijo za tisti jezik, ki je aktiven, ko je uporabljen ukaz `hyphenation`. To pomeni, da če ukaz `hyphenation` stoji v preambuli, potem so to namigi za deljenje besed v angleščini. Če pa ukaz uporabimo za `\begin{document}` in uporabljamo kakšen paket za podporo svojega jezika kot npr. `babel`, potem bodo namigi za deljenje aktivirani v okviru paketa `babel`.

Spodnji primer omogoča, da se beseda »hyphenation« deli in prav tako »Hyphenation«, preprečuje pa vsakršno deljenje besed »FORTRAN«, »Fortran« in »fortran«. V argumentu ukaza ni dovoljeno uporabljati posebnih znakov, posamezne besede pa ločimo s presledki.

```
\hyphenation{FORTRAN Hy-phen-a-tion}
```

Namesto `hyphenation` lahko na samem mestu, kjer potrebujemo deljenje besede, uporabimo ukaze `\-`, ki označujejo potencialna mesta deljenja v besedi. Ta ukaz je še posebno uporaben za besede, ki vsebujejo posebne znake (npr. znake z akcenti), saj \LaTeX ne zna avtomatično deliti besed, ki vsebujejo posebne znake.

1. Čeprav \LaTeX izpiše opozorilo (Overfull hbox) in izpiše predolgo vrstico, takih vrstic ni preprosto poiskati. Če pri ukazu `\documentclass` označimo opcijo `draft`, potem bodo predolge vrstice označene z debelo črno črto na desnem robu.

I think this is: su\~per\~cal\~%
i\~frag\~i\~lis\~tic\~ex\~pi\~%
al\~i\~do\~cious

I think this is: supercalifragilisticexpiali-
docious

Če ne želimo deljenja besede oziroma želimo, da več besed stoji skupaj v eni vrstici brez preloma vrstice, uporabimo ukaz

`\mbox{text}`

Ukaz povzroči, da argument stoji skupaj v eni vrstici ne glede na okoliščine.

Moja telefonska številka je spremenjena. Nova številka je `\mbox{0116 291 2319}`.

Moja telefonska številka je spremenjena. Nova številka je 0116 291 2319.

Parameter `\mbox{\emph{filename}}` naj vsebuje ime datoteke.

Parameter *filename* naj vsebuje ime datoteke.

Ukaz `\fbox` je podoben `\mbox`, dodatno pa nariše še vidni pravokotnik okrog vsebine.

2.3 Že pripravljeni nizi

V nekaterih primerih na prejšnjih straneh ste lahko opazili nekaj zelo preprostih \LaTeX ovih ukazov, ki izpišejo tekstovni niz z določeno vsebino:

Ukaz	Primer	Opis
<code>\today</code>	8. maj 2008	Trenutni datum v izbranem jeziku
<code>\TeX</code>	\TeX	Ime vašega priljubljenega programa
<code>\LaTeX</code>	\LaTeX	Glavna zadeva
<code>\LaTeXe</code>	$\LaTeX 2_{\epsilon}$	Trenutna inkarnacija \LaTeX a

2.4 Specialni znaki in simboli

2.4.1 Narekovaji

Za narekovaje se *ne* uporablja " kot na pisalnem stroju. Pri tisku poznajo posebne začetne in zaključne narekovaje. V \LaTeX u uporabimo dva znaka ' za začetne narekovaje in dva znaka ' (apostrof) za zaključne narekovaje. Za enojne narekovaje uporabimo po en znak ' in '.

‘‘Prosim, pritisnite ‘x’ tipko.’’

“Prosim, pritisnite ‘x’ tipko.”

Seveda to velja le za narekovaje v angleščini. Za pravilen zapis narekovajev v slovenščini si poglejte razdelek 2.5.1.

2.4.2 Pomišljaji in vezaji

L^AT_EX pozna štiri vrste vezajev. Do treh lahko pridemo tako, da vnesemo od enega do tri zaporedne pomišljaje. Četrty znak v bistvu sploh ni vezaj, saj gre za matematični simbol minus:

```
daughter-in-law, X-rated\\
pages 13--67\\
yes---or no? \\
$0$, $1$ and $-1$
```

```
daughter-in-law, X-rated
pages 13-67
yes—or no?
0, 1 and -1
```

Imena so naslednja: ‘-’ vezaj, ‘—’ pomišljaj, ‘—’ dvodelni pomišljaj in ‘-’ minus.

2.4.3 Tilda (~)

Znak, ki se pogosto pojavlja v internetnih naslovih, je tilda. V L^AT_EXu jo lahko dobimo z `\~`, vendar rezultat `~` ni povsem to, kar želimo. Namesto tega lahko poskusite:

```
http://www.rich.edu/~{bush} \\
http://www.clever.edu/$\sim$demo
```

```
http://www.rich.edu/~bush
http://www.clever.edu/~demo
```

2.4.4 Znak za stopinje (°)

Kako v L^AT_EXu izpišemo znak za stopinje, prikazuje naslednji primer.

```
Pri $-30\,\sim\mathrm{C}$
se bo kmalu začela superprevodnost.
```

```
Pri -30 °C se bo kmalu začela superpre-
vodnost.
```

Uporabimo lahko tudi paket `textcomp`, kjer je znak za stopinje na voljo kot `\textcelsius`.

2.4.5 Znak za valuto evro (€)

Za pisanje denarnih zneskov potrebujemo znak za evro. Ta je vključen v številne pisave. Če v preambuli dokumenta naložimo paket `textcomp` z

ukazom

```
\usepackage{textcomp}
```

imamo potem na voljo ukaz

```
\texteuro
```

za znak za evro.

Če v naši pisavi ni simbola za evro oziroma je na voljo, pa nam ni vseč, imamo na voljo še dve dodatni možnosti.

Prva je paket `eurosym`. V njem je na voljo uradni simbol za evro:

```
\usepackage[official]{eurosym}
```

Če bi raje uporabili simbol za evro, ki bi se ujemal z izbrano pisavo, lahko uporabimo opcijo `gen` namesto opcije `official`.

V paketu `marvosym` je na voljo mnogo različnih simbolov, vključno z znakom za evro, ki ima ime `\EURtm`. Slabost tega paketa je, da ne podpira nagnjenih in krepkih variant znaka za evro..

Tabela 2.1: Polna vreča znakov za evro

LM+textcomp	<code>\texteuro</code>	€	€	€
eurosym	<code>\euro</code>	€	€	€
[gen]eurosym	<code>\euro</code>	€	€	€
marvosym	<code>\EURtm</code>	◻01	◻01	◻01

2.4.6 Tri pike (...)

Na pisalnem stroju vejica in pika zavzameta enako velik prostor kot katerikoli drug znak. V knjigah ta znaka zavzameta le malo prostora in stojita zelo blizu predhodnih črk. Zaradi tega ne moremo vnesti tri pike tako, da bi le vnesli tri pike, saj bi bili presledki napačni. Za tri pike imamo poseben ukaz

```
\ldots
```

Ne tako ... ampak tako:\\
New York, Tokio, Budimpešta, \ldots

Ne tako ... ampak tako:
New York, Tokio, Budimpešta, ...

2.4.7 Ligature

Nekatere kombinacije črk se ne stavijo s postavljanjem različnih črk druge k drugi, temveč se uporabi posebne simbole.

ff fi fl ffi... namesto ff fi fl ffi ...

Te t.i. ligature lahko preprečimo, če med ustrezni dve črki vrinemo `\mbox{}`. To je lahko pomembno, če je beseda sestavljena iz dveh samostojnih besed.

Not shelfful\\
but shelf\mbox{ }ful

Not shelfful
but shelfful

2.4.8 Akcenti in posebni znaki

\LaTeX podpira uporabo akcentov in posebnih znakov, ki jih poznajo ne-angleški jeziki. Tabela 2.2 prikazuje vse možne akcente, s katerimi lahko opremimo črko o. Seveda to deluje tudi za ostale črke.

Če želimo postaviti akcent na vrh črke i ali j, potem moramo najprej odstraniti piko. To naredimo z ukazom `\i` oziroma `\j`.

H[^]otel, na^{\i} ve, ^{\'e}l^{\'e}ve,\\
sm^{\o} rrebr^{\o} d, !^{\'e}Se^{\~}norita!,\\
Sch^{\o}nbrunner Schlo^{\ss}
Stra^{\ss} e,
^{\v{s}}^{\v{c}}epec soli

Hôtel, naïve, élève,
smørrebrød, ¡Señorita!,
Schönbrunner Schloß Straße, ščepec soli

2.5 Podpora drugih jezikov

Če pišemo besedila v jeziku, ki ni angleščina, potem moramo \LaTeX na treh mestih pravilno nastaviti:

1. Vsi nizi, ki se avtomatično generirajo² morajo biti prilagojeni novemu jeziku. Za večino jezikov, med njimi tudi za slovenščino, lahko to dosežemo z uporabo paketa `babel`, ki ga je napisal Johannes Braams.

2. npr. Table of Contents, List of Figures, ...

2. \LaTeX mora poznati pravila za deljenje besed za novi jezik. Za večino jezikov (tudi za slovenščino) je to že vključeno v paket `babel`, le pri namestitvi je \LaTeX u potrebno povedati, da naj jih uporablja, kar je odvisno od operacijskega sistema lahko malce bolj zapleteno opravilo. Če podpore za jezik ni, je potrebno ročno zgraditi datoteko pravil za deljenje.
3. Potrebno je nastaviti specifična pravila za stavljenje teksta. Npr., v francoščini mora obvezno stati presledek pred vsakim dvopičjem (:).

Če je sistem že pravilno nastavljen, potem lahko paket `babel` aktiviramo z ukazom

```
\usepackage[jezik]{babel}
```

za ukazom `\documentclass`. Argument *jezik* je ime jezika, ki ga podpira sistem. V primeru slovenščine je to `slovene`. Pri vsakem klicu prevajalnika se izpiše seznam jezikov, ki so vgrajeni v naše \LaTeX okolje. Babel avtomatično vključi ustrezna pravila za deljenje za izbrani jezik. Če \LaTeX ne podpira deljenja v izbranem jeziku, bo `babel` še vedno delal, toda deljenje besed bo ali izključeno ali pa nepravilno, kar ima lahko negativne posledice na sam videz prevedenega dokumenta.

Pri nekaterih jezikih `babel` tudi definira nove ukaze, ki poenostavijo vnos posebnih znakov oziroma lokalnih črk. V slovenščini tako lahko šumnike pišemo kot `"c`, `"s`, `"z`, `"C`, `"S`, `"Z` namesto `\v{c}`, `\v{s}`, `\v{z}`, `\v{C}`, `\v{S}`, `\v{Z}`. V vseh novejših različicah \LaTeX a lahko šumnike vnašamo tudi kot `č`, `š`, `ž`, `Č`, `Š`, `Ž`, podrobnosti najdete v nadaljevanju.

Tabela 2.2: Akcenti in posebni znaki.

<code>ò</code>	<code>\'o</code>	<code>ó</code>	<code>\'o</code>	<code>ô</code>	<code>\^o</code>	<code>õ</code>	<code>\~o</code>
<code>ō</code>	<code>\=o</code>	<code>ò</code>	<code>\.o</code>	<code>ö</code>	<code>\"o</code>	<code>ç</code>	<code>\c c</code>
<code>ö</code>	<code>\u o</code>	<code>ő</code>	<code>\v o</code>	<code>ő</code>	<code>\H o</code>	<code>ç</code>	<code>\c o</code>
<code>ø</code>	<code>\d o</code>	<code>ø</code>	<code>\b o</code>	<code>ö</code>	<code>\t oo</code>		
<code>œ</code>	<code>\oe</code>	<code>Œ</code>	<code>\OE</code>	<code>æ</code>	<code>\ae</code>	<code>Æ</code>	<code>\AE</code>
<code>å</code>	<code>\aa</code>	<code>Å</code>	<code>\AA</code>				
<code>ø</code>	<code>\o</code>	<code>Ø</code>	<code>\O</code>	<code>ł</code>	<code>\l</code>	<code>Ł</code>	<code>\L</code>
<code>ı</code>	<code>\i</code>	<code>ı</code>	<code>\j</code>	<code>ı</code>	<code>!'</code>	<code>ı</code>	<code>?'</code>

Če pri klicu paketa `babel` navedemo več jezikov

```
\usepackage[jezikA,jezikB]{babel}
```

potem bo aktiven zadnji jezik iz seznama (v zgornjem primeru `jezikB`). Z ukazom

```
\selectlanguage{jezikA}
```

lahko spremenimo aktivni jezik.

Večina modernih operacijskih sistemih dopušča, da posebne znake nacionalne abecede vnašamo kar direktno s tipkovnico. Za obdelavo različnih vhodnih kodnih tabel, ki jih potrebujemo za različne skupine jezikov in različna računalniška okolja, \LaTeX uporablja paket `inputenc`, ki ga vključimo z ukazom

```
\usepackage[kodna tabela]{inputenc}
```

Pri uporabi tega paketa moramo razmišljati o tem, da zaradi uporabe različnih kodnih tabel na različnih operacijskih sistemih, dokument s takim kodiranjem mogoče ne bo možno predelati na drugem računalniku. Tipičen primer so šumniki, ki imajo pod Okni na PC drugačne kode kot pa na Unixu. Naslednja tabela prikazuje nekatera imena kodnih tabel, ki jih lahko uporabimo kot argument v paketu `inputenc`:³

operacijski sistem	kodna tabela
Mac	<code>applemac</code>
Unix	<code>latin1</code>
Windows	<code>ansinew</code>
OS/2	<code>cp850</code>
MS-DOS (šumniki)	<code>cp852</code>
Windows (šumniki)	<code>cp1250</code>
Unix (šumniki)	<code>latin2</code>

Če imate večjezični dokument, v katerem se iste kode znakov uporabljajo v različnih jezikih za različne črke, je mogoče bolje, da preklopite na kodiranje unicode z uporabo paketa `ucs`. Ukaza

```
\usepackage{ucs}
\usepackage[utf8x]{inputenc}
```

vam omogočata kreiranje \LaTeX ove vhodne datoteke v kodiranju `utf8x`, kjer je vsak znak predstavljen z najmanj enim in največ štirimi bajti.

3. Več o podprtih vhodnih kodiranjih za jezike, ki uporabljajo latinico in cirilico, lahko najdete v v dokumentaciji `inputenc.dtx` oziroma `cyinpenc.dtx`. V razdelku 4.6 preberite, kako se pripravi navodila za pakete iz `.dtx` datotek.

Samo kodiranje pisave je druga zadeva. Ta pove, na katerem položaju v \TeX ovih pisavah je shranjen posamezni znak. Več vhodnih kodiranj se lahko preslika na isto kodiranje pisave, kar zmanjša število potrebnih pisav. Za kodiranje pisav skrbi paket `fontenc`:

```
\usepackage[kodiranje]{fontenc}
```

kjer je *kodiranje* izbrano kodiranje pisave. Možno je naložiti več kodiranj pisav naenkrat.

Privzeto \TeX ovo kodiranje pisave je OT1, kar ustreza originalni Computer Modern pisavi. Ta vsebuje le 128 znakov 7-bitnega ASCII nabora znakov. Ko potrebujemo znake z akcenti, jih \TeX naredi tako, da kombinira normalno črko z ustreznim akcentom. Čeprav rezultat zglada sijajno, pa tak pristop preprečuje avtomatično deljenje besed, ki vsebujejo znake z akcenti. Poleg tega nekaterih znakov v latinici ni možno sestaviti s kombiniranjem akcenta in normalne črke, da grških črk in cirilice sploh ne omenjamo.

Za odpravo te pomanjkljivosti so razvili nekaj 8 bitnih pisav, ki so podobne Computer Modern (CM). Tako *Extended Cork* (EC) pisave v T1 kodiranju vsebujejo črke in ločila za večino evropskih jezikov, ki uporabljajo latinico. Pisave LH vsebujejo črke za pisavo dokumentov v cirilici. Ker je znakov v cirilici veliko, so urejeni v štiri kodiranja—T2A, T2B, T2C in X2.⁴ Skupina CB vsebujejo pisave v LGR kodiranju za sestavljanje grških tekstov.

Z uporabo teh pisav lahko izboljšamo deljenje besed v neangleških besedilih. Še ena prednost je tudi, da imamo na voljo CM pisave vseh debelin, nagibov in poljubne optične povečave.

2.5.1 Podpora za slovenščino

Tu je nekaj kratkih napotkov za pisanje slovenskih dokumentov. Najprej je potrebno naložiti paket `babel` in izbrati slovenščino.

```
\usepackage[slovene]{babel}
```

To vključi slovenska pravila za deljenje besed, če je \LaTeX pravilno nameščen (če ni, pogledjte navodila [19].) Teksti, ki jih \TeX avtomatično generira, so prevedeni. Tako npr. »Chapter« postane »Poglavje«. Poleg tega imamo sedaj na voljo nekaj novih ukazov za lažji zapis slovenskih tekstov, ki so prikazani v tabeli 2.3.

Namesto "c", "s", ..., lahko do šumnikov pridemo tudi tako, da v vhodno datoteko kar direktno pišemo šumnike, če pravilno nastavimo vhodno kodno tabelo. Se pa moramo zavedati, da so potem takšni vhodni dokumenti prenosljivi le znotraj istega operacijskega sistema. Za rezultate to

4. Seznam jezikov, ki jih podpira vsako izmed teh kodiranj, lahko najdete v [11].

ni pomembno, saj so DVI datoteke neodvisne od kodne tabele v vhodnih podatkih.

Priporočljivo je tudi vedno uporabljati pisave EC, zaradi deljenja besed s šumniki.

Kratek dokument v slovenščini (pomebna je le preambula) je prikazan na sliki 2.1. Pomembno je tudi, da v ukazu `documentclass` uporabimo opcijo `a4paper`, saj je privzeta velikost papirja `letter`, medtem ko pri nas uporabljamo papir A4.

2.6 Razmiki med besedami

Da bi bilo besedilo tudi na desni strani poravnano, \LaTeX vstavlja med besede različno široke razmike. Na koncu stavka vstavi malo večji presledek, da je tekst lažje berljiv. \LaTeX predpostavlja, da se stavek konča s piko, vprašajem ali klicajem. Če pika sledi veliki črki, potem to ne vzame za konec stavka, saj se velike črke in pika normalno pojavljajo v okrajšavah.

Vsako odstopanje od teh pravil mora avtor sam izrecno navesti. Znak `\` pred presledkom naredi presledek, ki ga \LaTeX ne bo raztegnil. Znak `~` se prevede v presledek, ki ga ni moč povečati, na tem mestu pa tudi ni moč zlomiti vrstice. Ukaz `\@` pred piko pomeni, da ta pika končuje stavek, ne

Tabela 2.3: Posebni znaki v slovenščini.

"c	č	"C	Č
"s	š	"S	Š
"z	ž	"Z	Ž
"‘	”	"’	“
"<	«	">	»

```

\documentclass[a4paper]{article}
\usepackage[slovene]{babel}
\usepackage[cp1250]{inputenc}
\usepackage[T1]{fontenc}
\begin{document}
To je zelo zelo preprosto besedilo v slovenščini.
\end{document}

```

Slika 2.1: Primer vhodne \LaTeX datoteke v slovenščini.

glede na to, da je pred njo velika črka.

```
Mr.~Smith was happy to see her\\
cf.~Fig.~5\\
I like BASIC\@. What about you?
```

```
Mr. Smith was happy to see her
cf. Fig. 5
I like BASIC. What about you?
```

Dodatne presledke za vejico lahko izključimo z ukazom

```
\frenchspacing
```

ki pove \LaTeX u, da naj za vejico ne vstavlja večji presledek kot za normalno črko. To je zelo pogosto v neangleških jezikih, razen v seznamu literature. Če uporabimo ukaz `\frenchspacing`, potem ukaza `\@` ne potrebujemo.

2.7 Naslovi, poglavja in razdelki

Da se bo bralec lažje znašel v vašem dokumentu, se ga spodobi razdeliti na logične enote, kot so poglavja, razdelki in podrazdelki. \LaTeX ima na voljo posebne ukaze, kjer kot argument navedemo naslov enote. Paziti moramo na to, da jih uporabljamo v pravilnem vrstnem redu.

Naslednji ukazi za logične enote so na voljo v razredu `article`:

```
\section{...}      \paragraph{...}
\subsection{...}   \subparagraph{...}
\subsubsection{...}
```

Kadar želimo razdeliti dokument na dele, ki na ne vplivajo na oštevilčenje razdelkov oziroma poglavij, uporabimo

```
\part{...}
```

V razredih `report` in `book` je na voljo še dodatni višji nivo za deljenje dokumentov:

```
\chapter{...}
```

Ker razred `article` ne pozna poglavij, je zelo preprosto iz posameznih člankov narediti poglavja knjige. Za presledke med enotami, njihovo oštevilčenje in velikost pisave avtomatično skrbi \LaTeX .

Dva ukaza za logično delitev besedila sta posebna:

- Ukaz `\part` ne vpliva na oštevilčenje poglavij.
- Ukaz `\appendix` je brez argumenta. Vse kar naredi je, da oštevilčenje poglavij spremeni v črke.⁵

5. V razredu `article` spremeni oštevilčenje razdelkov.

\LaTeX sam naredi kazalo iz naslovov logičnih enot, številke strani pa vzame iz prejšnjega prevajanja dokumenta. Ukaz

```
\tableofcontents
```

na mestu uporabe pripravi kazalo. Zaradi morebitnih sprememb strani je potrebno vhodni dokument prevesti z \LaTeX om dvakrat, da bodo strani v kazalu pravilne, v nekaterih primerih pa so potrebna tudi tri prevajanja. Pri prevajanju \LaTeX sam opozori, kdaj je priporočljivo dokument še enkrat prevesti.

Vsi ukazi za logične enote imajo tudi različico »z zvezdico«. Dobimo jih tako, da ustreznemu ukazu na koncu dodamo zvezdico *. Tak ukaz pripravi naslov enote, vendar se ta naslov ne pojavi v kazalu in ni oštevilčen. Ukaz `\section{Pomoč}` lahko tako spremenimo v `\section*{Pomoč}`, če želimo neoštevilčeni razdelek, ki se ne bo pojavil v kazalu.

Normalno se naslovi enot pojavijo v kazalu v takšni obliki kot so vnešeni v tekst. V nekaterih primerih to ni možno, saj je lahko naslov predolg za eno vrstico v kazalu. V takšnem primeru lahko naslov, ki naj se izpiše v kazalu, podamo kot dodatni argument pred dejanskim naslovom

```
\chapter[Kratek naslov za kazalo]{Dolg
in še posebno dolgočasen naslov,
ki se izpiše na začetku poglavja}
```

Naslov celotnega dokumenta se naredi z ukazom

```
\maketitle
```

Vsebino naslova je potrebno definirati z ukazoma

```
\title{...}, \author{...} in po želji tudi z ukazom \date{...}
```

še preden kličemo ukaz `\maketitle`. Kot argument ukaza `\author` lahko podamo več imen, ki jih ločimo z ukazom `\and`.

Primer uporabe nekaterih izmed naštetih ukazov je na sliki 1.2 na strani 7.

Poleg že omenjenih ukazov $\LaTeX 2_{\epsilon}$ pozna še nekaj dodatnih ukazov, ki jih lahko uporabimo v razredu `book`. Primerni so za razdelitev publikacije na tri dele in spremenijo naslove poglavij in oštevilčenje strani, kot je to v navadi v knjigi:

`\frontmatter` naj bi bil čisto prvi ukaz za `\begin{document}`. Ta ukaz spremeni oštevilčenje strani v rimske številke. Na začetku knjige uporabljamo neoštevilčene logične enote, kot npr. `\chapter*{Predgovor}`, ki ne nastopajo v kazalu, strani pa so oštevilčene z rimskimi številkami.

`\mainmatter` uporabimo takoj za prvim poglavjem v knjigi (za poglavjem z rimskimi številkami), ukaz pa spremeni oštevilčenje strani na arabske številke in postavi števec strani na začetek.

`\appendix` Označuje začetek dodatka v knjigi. Od tu naprej so poglavja oštevilčena s črkami.

`\backmatter` vstavimo pred zadnjimi stvarmi v knjigi, kot sta seznam literature in stvarno kazalo, ukaz pa izklopi oštevilčenje logičnih enot. V preostalih standardnih razredih dokumentov ukaz nima nobenega učinka.

2.8 Sklicevanja

V knjigah, poročilih in člankih se velikokrat sklicujemo na slike, tabele in druge dele teksta. \LaTeX ima na voljo naslednje ukaze za sklicevanja

```
\label{marker}, \ref{marker} in \pageref{marker}
```

kjer je *marker* oznaka, ki jo izbere uporabnik in mora biti seveda enolična. \LaTeX zamenja `\ref` z oznako razdelka, podrazdelka, slike, tabele, ali izreka, kjer smo uporabili ustrezni ukaz `\label`. Ukaz `\pageref` izpiše številko strani, na kateri je med besedilom ukaz `\label`.⁶ Kot pri naslovi enot in straneh v kazalu, se tudi tu uporabijo oznake iz prejšnjega prevajanja besedila.

Sklic na ta podrazdelek

```
\label{sec:this} ima obliko:\>poglej podrazdelek~\ref{sec:this}
na strani ~\pageref{sec:this}."<
```

Sklic na ta podrazdelek ima obliko:
»poglej podrazdelek 2.8 na strani 31.«

2.9 Opombe

Z ukazom

```
\footnote{footnote text}
```

izpišemo opombo pod črto na dnu tekoče strani. Opombe vedno vstavljamo za besedo oziroma stavkom, na katerega se nanašajo. Opombe na del stavka je zato potrebno postaviti za piko ali vejico.⁷

6. Ti ukazi ne vedo, na kaj se sklicujejo. `\label` shrani le zadnjo avtomatično narejeno oznako.

7. Opombe odmikajo bralca od glavnega besedila. Res je, da vsi beremo opombe, ker smo radovedni, ampak ali ni bolje vsega, kar želimo povedati, vstaviti v glavni dokument?⁸

8. Pred dajanjem nasvetov bi bilo mogoče dobro pomesti pred lastnim pragom . . .

Opombe`\footnote{To je opomba.}` se v `\LaTeX{}`u pogosto uporabljajo.

Opombe^a se v `\LaTeX`u pogosto uporabljajo.

a. To je opomba.

2.10 Poudarjene besede

Če tekst pišemo na pisalnem stroju, potem pomembne besede poudarimo tako, da jih podčrtamo. Za podčrtanje je v `\LaTeX`u na voljo ukaz

```
\underline{text}
```

V natisnjenih knjigah so besede ponavadi poudarjene tako, da so zapisane s *poševno* pisavo. V `\LaTeX`u imamo za poudarjanje besed na voljo ukaz

```
\emph{text}
```

Rezultat ukaza je odvisen od konteksta, kjer ga uporabimo:

```
\emph{Če poudarimo
  tekst znotraj poudarjenega
  tekst, potem ga \LaTeX{}}
poudari z \emph{normalno}
pisavo.}
```

Če poudarimo tekst znotraj poudarjenega tekst, potem ga `\LaTeX` poudari z normalno pisavo.

Bodite pozorni na razliko med tem, ko `\LaTeX`u naročimo, da naj *poudari* tekst in tem, ko mu naročimo, da naj uporablja drugačno *pisavo*:

```
\textit{Tudi v poševni pisavi
  lahko \emph{poudarimo} tekst,}
\textsf{prav tako v \emph{gladki}
  (sans serif) pisavi,}
\texttt{in v pisavi
  \emph{pisalnega stroja}.}
```

Tudi v poševni pisavi lahko poudarimo tekst, prav tako v gladki (sans serif) pisavi, in v pisavi pisalnega stroja.

2.11 Okolja

Za izpis teksta posebne oblike (naštevanja, sredinsko poravnani tekst,...) so v `\LaTeX`u na voljo številna okolja. Okolje se začne z `\begin{ime okolja}`

in konča z `\end{ime okolja}`, vmes pa postavimo tekst, za katerega želimo poseben izpis:

```
\begin{okolje} tekst \end{okolje}
```

Tu je *okolje* naziv okolja. Okolja lahko gnezdimo enega v drugega vse dokler uporabljamo pravilni vrstni red, kot npr.

```
\begin{aaa}...\begin{bbb}...\end{bbb}...\end{aaa}
```

V naslednjih podrazdelkih so opisana vsa pomembnejša okolja.

2.11.1 Sezname

Okolje `itemize` je primerno za preproste sezname, kjer navajamo stvari, okolje `enumerate` je namenjeno za oštevilčene sezname, okolje `description` pa za sezname, kjer opisujemo zadeve.

```
\flushleft
\begin{enumerate}
\item Različna okolja lahko mešamo
po lastnem okusu:
\begin{itemize}
\item Toda to lahko postane smešno.
\item[-] To se začne s pomišljajem.
\end{itemize}
\item Zapomnite si:
\begin{description}
\item[Neumne] zadeve ne bodo
postale pametne, če so v
seznamu.
\item[Pametne] zadeve, za razliko,
pa lahko čudovito prikažemo
s seznamom.
\end{description}
\end{enumerate}
```

1. Različna okolja lahko mešamo po lastnem okusu:

- Toda to lahko postane smešno.
- To se začne s pomišljajem.

2. Zapomnite si:

Neumne zadeve ne bodo postale pametne, če so v seznamu.

Pametne zadeve, za razliko, pa lahko čudovito prikažemo s seznamom.

2.11.2 Levo, desno in sredinsko poravnan tekst

Okolji `flushleft` in `flushright` naredita odstavke, ki so levo oziroma desno poravnani. Tekst v okolju `center` je sredinsko poravnan. Če z ukazom `\\` ne povemo, kje so prelomi, bo \LaTeX sam določil prelome vrstic.

```
\begin{flushleft}
Ta tekst je \\ levo poravnan.
\LaTeX{} se ne trudi narediti
vrstic z enakimi dolžinami.
\end{flushleft}
```

Ta tekst je levo poravnan. \LaTeX se ne trudi narediti vrstic z enakimi dolžinami.

```
\begin{flushright}
Ta tekst je desno \\poravnan.
Kot prej tudi tu vrstice nimajo
enakih dolžin.
\end{flushright}
```

Ta tekst je desno
poravnan. Kot prej tudi tu vrstice
nimajo enakih dolžin.

```
\begin{center}
V središču\\sveta
\end{center}
```

V središču
sveta

2.11.3 Navedki in verzi

Okolje `quote` uporabljamo za navedke, pomembne fraze in primere.

```
Splošno tiskarsko pravilo za
dolžino vrstice je:
\begin{quote}
V povprečju ne sme imeti nobena
vrstica več kot 66 znakov.
\end{quote}
Zaradi tega imajo strani v
\LaTeX{}u privzete tako velike
robove in zato se v časopisih
uporablja več stolpcev.
```

Splošno tiskarsko pravilo za dolžino vrstice je:

V povprečju ne sme imeti no-
bena vrstica več kot 66 zna-
kov.

Zaradi tega imajo strani v \LaTeX u privzete tako velike robove in zato se v časopisih uporablja več stolpcev.

Dve podobni okolji sta `quotation` in `verse`. Okolje `quotation` uporabljamo za daljše navedke, ki so lahko sestavljeni iz več odstavkov, saj so v tem okolju odstavki zamaknjeni. Okolje `verse` pa je pomembno za pisanje pesmi, kjer so prelomi vrstic pomembni. Vrstice ločimo z ukazom `\\` na koncu vrstice, kitice pa s prazno vrstico.

```
Tole pesem najbrž vsi poznamo.
\begin{flushleft}
\begin{verse}
Barčica po morju plava,\\
drevesa se priklanjajo.\\
O le naprej, o le naprej,\\
dokler je še vetra kej.

Barčica po morju plava \ldots
\end{verse}
\end{flushleft}
```

Tole pesem najbrž vsi poznamo.

Barčica po morju plava,
drevesa se priklanjajo.
O le naprej, o le naprej,
dokler je še vetra kej.
Barčica po morju plava ...

2.11.4 Povzetek

V znanstvenih publikacijah je običajno na začetku povzetek (abstrakt), v katerem je kratek pregled vsebine dokumenta. \LaTeX ima v ta namen na voljo okolje `abstract`. Okolje `abstract` uporabljamo v dokumentih, ki so napisani v razredu `article`.

```
\begin{abstract}
To je povzetek.
\end{abstract}
```

To je povzetek.

2.11.5 Dobesedni izpis

Tekst med `\begin{verbatim}` in `\end{verbatim}` bo natisnjen tako, kot da bi ga natisnili s pisalnim strojem, z vsemi prelomi vrst in presledki vred in brez izvajanja morebitnih \LaTeX ovih ukazov.

Znotraj odstavka lahko podobno dosežemo z

```
\verb+text+
```

Črka `+` je le ena možnost, v resnici lahko uporabimo poljuben par enakih znakov za označitev začetka in konca, razen črk, znaka `*` ali presledka. Veliko \LaTeX zgledov v tej knjigi je natisnjenih ravno s tem ukazom.

Ukaz `\verb|\ldots|` naredi `\ldots`

```
\begin{verbatim}
10 PRINT "HELLO WORLD ";
20 GOTO 10
\end{verbatim}
```

Ukaz `\ldots` naredi ...

```
10 PRINT "HELLO WORLD ";
20 GOTO 10
```

```
\begin{verbatim*}
različica z zvezdico
okolja      verbatim
poudari    presledke
v besedilu
\end{verbatim*}
```

```
različica_z_zvezdico
okolja_verbatim
poudari_presledke
v_besedilu
```

Podobno lahko tudi ukazu `\verb` dodamo na konec zvezdico:

```
\verb*|kot kaže ta primer :-)| kot_kaže_uta_primer_:-)|
```

Okolja `verbatim` in ukaza `\verb` ne moremo uporabljati znotraj argumentov drugih ukazov.

2.11.6 Tabele

Okolje `tabular` uporabljamo za stavljenje lepih tabel, ki imajo lahko tudi vodoravne in navpične črte. Širino stolpcev \LaTeX določi avtomatično (če ni podano drugače).

Argument *stolpci* v ukazu

```
\begin{tabular}[pos]{table spec}
```

določa število stolpcev in poravnavanje v vsakem stolpcu. Pri tem `l`, `r` in `c` pomenijo stolpec z levo, desno oziroma sredinsko poravnanim tekstom; `p{širina}` pomeni stolpec z dano širino in z obojestransko poravnanim tekstom, kjer se predolg tekst avtomatično zlomi v več vrstic, znak `|` pa pomeni navpično črto.

Če je tekst znotraj stolpca preširok, ga \LaTeX ne bo avtomatično prelomil v novo vrstico. Z ukazom `p{širina}` definiramo posebno vrsto stolpca, kjer se tekst lomi kot v normalnem odstavku,

Argument *pos* določa vertikalno poravnavo tabele relativno glede na dno besedila, ki jo obkroža. Z vrednostmi `t`, `b` oziroma `c` nastavimo poravnavo tabele glede na vrh, dno ali sredino.

V okviru okolja `tabular` znak `&` pomeni konec stolpca in začetek naslednjega, `\\` začneja novo vrstico, z ukazom `\hline` pa vstavimo vodoravno črto. Če želimo vodoravno črto, ki se razteza le čez nekaj stolpcev, uporabimo ukaz `\cline{i-j}`, kjer sta *i* in *j* indeksa stolpcev kjer se črta začne oziroma konča.

```
\begin{tabular}{|r|l|}
\hline
7C0 & hexadecimal \\
3700 & octal \\ \cline{2-2}
11111000000 & binary \\
\hline \hline
1984 & decimal \\
\hline
\end{tabular}
```

7C0	hexadecimal
3700	octal
11111000000	binary
1984	decimal

```
\begin{tabular}{|p{4.7cm}|}
\hline
Dobrodošli v škatlastem odstavku.
Upam, da ste zaenkrat zadovoljni
s tem, kar berete.\\
\hline
\end{tabular}
```

Dobrodošli v škatlastem odstavku. Upam, da ste zaenkrat zadovoljni s tem, kar berete.

Ukaz `@{besedilo}` v vsaki vrstici med stolpcema namesto praznega prostora izpiše besedilo med zavritimi oklepaji. Zgled za uporabo je v nadal-

jevanju predstavljen kot rešitev problema poravnavanja števil na decimalni piki. Druga možna uporaba je, kadar želimo tabelo brez začetnega praznega prostora v prvem stolpcu oziroma končnega praznega prostora v zadnjem stolpcu. V ta namen uporabimo `@{}`.

```
\begin{tabular}{@{} l @{}}
\hline
brez presledka
na začetku in na koncu\\
\hline
\end{tabular}
```

brez presledka na začetku in na koncu

```
\begin{tabular}{l}
\hline
presledek na začetku in koncu\\
\hline
\end{tabular}
```

presledek na začetku in koncu

Ker ni na voljo nobenega vgrajenega orodja, ki bi vsebino stolpcev z decimalnimi števili poravnalo na decimalni piki,⁹ lahko malo »goljufamo« in to naredimo z uporabo dveh stolpcev: desno poravnane z celim delom števila in levo poravnane z decimalnim delom. Ukaz `@{.}` v `\begin{tabular}` zamenja normalni presledek med dvema stolpcema s piko '.' in dobimo videz enojnega stolpca, kjer so podatki poravnani po decimalni piki. Pri vnosu decimalnih števil v takšno tabelo ne smemo pozabiti decimalne pike zamenjati z znakom za ločevanje stolpcev &. Naslov stolpca nad stolpcem, ki je v resnici sestavljen iz dveh stolpcev, dobimo z ukazom `\multicolumn`.

```
\begin{tabular}{c r @{.} l}
izraz s številom $\pi$ & & \\
\multicolumn{2}{c}{vrednost} \\
\hline
$\pi$ & 3.1416 & \\
$\pi^\pi$ & 36.46 & \\
$(\pi^\pi)^\pi$ & 80662.7 & \\
\end{tabular}
```

izraz s številom π	vrednost
π	3.1416
π^π	36.46
$(\pi^\pi)^\pi$	80662.7

```
\begin{tabular}{|c|c|}
\hline
\multicolumn{2}{|c|}{Ene} \\
\hline
Mene & Muh! \\
\hline
\end{tabular}
```

Ene	
Mene	Muh!

9. Poravnavo decimalnih števil na piki omogoča dodatni paket `dcolumn`.

Vsebina okolja `tabular` vedno ostane na eni strani, saj v tem okolju ni preloma strani. Če potrebujemo dolge tabele, ki se raztezajo čez več strani, si lahko pomagamo s paketom `longtable`.

2.12 Plavajoči objekti

Večina sodobnih publikacij vsebuje veliko slik in tabel. Ti objekti potrebujejo posebno obdelavo, saj jih ni možno deliti med stranmi. En način bi bil, da bi vedno, ko je tabela ali slika prevelika, da bi se jo še dalo spraviti na tekočo stran, naredili novo stran. Omenjeni pristop bi pustil strani delno nezapolnjene, to pa bi zgledalo grdo.

Rešitev je, da sliko oziroma tabelo, ki je premajhna za tekočo stran, pustimo za naslednjo stran, prostor na tekoči strani pa zapolnimo z naslednjim tekstom. Tako si lahko predstavljamo slike in tabele kot telesa, ki plavajo med tekstom, postavimo pa jih na prvo naslednjo stran, na kateri imamo za objekt dovolj prostora. \LaTeX ponuja dve okolji za plavajoče objekte: `table` za tabele in `figure` za slike. Da lahko ti dve okolji čim bolj izkoristimo, moramo najprej vsaj približno razumeti, kako \LaTeX interno vodi plavajoče objekte. V nasprotnem primeru lahko plavajoči objekti postanejo glavni vzrok frustracij, saj jih \LaTeX nikoli ne postavi tam, kjer bi mi to želeli.

Najprej si pogledajmo ukaze, ki jim imamo v \LaTeX u za plavajoče objekte:

Vse kar je znotraj okolja `figure` oziroma `table` se obravnava kot en plavajoči objekt. Obe okolji poznata še opsijske parametre v obliki

```
\begin{figure}[lega] ali \begin{table}[lega]
```

ki določajo lego objekta. Parameter *lega* pove \LaTeX u, kam je dovoljeno postaviti objekt. Vrednost parametra je niz sestavljen iz ene ali več črk, ki označujejo dovoljene položaje. Možne oznake so navedene v tabeli 2.4.

Tabela 2.4: Oznake za dovoljene položaje plavajočega objekta.

Oznaka	Objekt lahko stoji ...
<code>h</code>	<i>tukaj</i> na mestu v tekstu, kjer je vstavljen. To pride v poštev za majhne plavajoče objekte.
<code>t</code>	na <i>vrhu</i> strani.
<code>b</code>	na <i>dnu</i> strani.
<code>p</code>	na posebni <i>strani</i> , ki vsebuje le plavajoče objekte.
<code>!</code>	brez upoštevanja večine vgrajenih parametrov ^a , ki bi lahko preprečile, da je objekt postavljen na to mesto.

^a. Kot npr. maksimalno število plavajočih objektov na strani.

Tabela se lahko začne npr. z naslednjo vrstico

```
\begin{table}[!hbp]
```

Niz `[!hbp]` dovoljuje \LaTeX u da postavi to tabelo točno na to mesto (`h`), na dno strani (`b`) ali pa na posebno stran, na kateri so le plavajoči objekti (`p`), to vse pa lahko naredi tudi, če rezultat ni najboljši (`!`). Če ne podamo opsijskega argumenta z lego, potem standardni razredi privzamejo lego `[tbp]`.

\LaTeX bo vsak plavajoči objekt, ki ga sreča, postavil na tak položaj, kot ga je predpisal avtor. Če objekta ne more postaviti na tekočo stran, se prenese v ustrezno čakalno vrsto (ena za tabele in ena za slike)¹⁰. Ko se začne nova stran, \LaTeX najprej preveri, če lahko celo stran zapolni z objekti iz obeh čakalnih vrst in tako naredi posebno stran le s tabelami in s slikami. Če to ni možno, potem v vsaki vrsti prvi objekt obravnava tako, kot da bi se ravnokar pojavil v besedilu in ga poskusi postaviti na ustrezno mesto (razen lege `h`, ki ni več možna). Vsak nov objekt, ki se pojavi v tekstu, gre v ustrezno čakalno vrsto. \LaTeX ves čas vzdržuje vrstni red objektov v takšnem vrstnem redu, kot se pojavijo v dokumentu. Zaradi tega slika, ki je ni možno postaviti na predpisano mesto, porine vse preostale slike proti koncu dokumenta. Torej:

Če \LaTeX ne razporedi plavajočih objektov po željah, potem je pogosto vzrok v tem, da je en plavajoči objekt zamašil eno izmed dveh čakalnih vrst.

Čeprav je možno pri določanju lege objekta uporabiti le en znak, to povzroča težave. Če je objekt prevelik za prostor, ki je na voljo, potem se zatakne in blokira plavajoče objekte, ki mu sledijo. Zaradi tega ne smemo za lego nikoli uporabiti le `[h]`, saj je to zelo slabo; tako zanič, da novejša različica \LaTeX a pri prevajanju `[h]` avtomatično spremeni v `[ht]`.

Sedaj, ko smo obdelali težavni del, lahko omenimo še nekaj stvari, ki se tičejo okolij `table` in `figure`. Z ukazom

```
\caption{pojasnilo}
```

lahko definiramo pojasnilo za objekt, kjer za oštevilčenje in niz »Figure« oziroma »Table« (v slovenščini »Slika« in »Tabela«) poskrbi \LaTeX .

```
\listoffigures in \listoftables
```

delujeta podobno kot `\tableofcontents`, izpišeta pa kazalo slik oziroma tabel. V teh kazalih se še enkrat ponovijo celotna pojasnila. Če uporablj

10. Vrsti sta tipa fifo - 'first in first out', oziroma, objekt, ki prej pride, gre tudi prej ven.

jamo dolga pojasnila, lahko podobno kot pri logičnih enotah kot dodatni parameter v oglatih oklepajih zapišemo kratko pojasnilo za kazalo.

```
\caption[Kratko]{Dooooooooooooo1111gooo}
```

S pomočjo ukazov `\label` in `\ref` se lahko v besedilu sklicujemo na plavajoči objekt.

Naslednji primer nariše kvadrat in ga vstavi v dokument. To lahko npr. naredimo, če v besedilu želimo rezervirati prostor za bodoče slike.

```
Figure~\ref{white} is an example of Pop-Art.
\begin{figure}[!hbp]
\makebox[\textwidth]{\framebox[5cm]{\rule{0pt}{5cm}}}
\caption{Five by Five in Centimetres.} \label{white}
\end{figure}
```

V zgornjem primeru se \LaTeX trudi *zares trudi* (!) postaviti sliko *točno* na to mesto (h).¹¹ Če to ne gre, poskuša postaviti sliko na dno strani. Ko ne uspe slike postaviti na tekočo stran, ugotavlja naprej, če je možno narediti stran s samimi plavajočimi objekti na naslednji strani. Če za takšno posebno stran ni dovolj objektov, potem \LaTeX začne novo stran in spet obravnava sliko, kot da bi se ravnokar pojavila v besedilu.

Pod določenimi pogoji je bolje uporabljati ukaza

`\clearpage` ali celo `\cleardoublepage`

Ukaza naročata \LaTeX u, da takoj razmesti vse preostale plavajoče objekte in nato začne novo stran. Razlika med ukazoma je, da se `\cleardoublepage` vedno nadaljuje na novi desni strani.

2.13 Zaščita krhkih ukazov

Tekst, ki je v argumentu nekaterih ukazov, kot npr. `\caption` ali `\section`, se lahko v dokumentu pojavi več kot enkrat (npr. enkrat kot naslov med tekstom in drugič v kazalu). Nekateri ukazi ne delujejo, če jih uporabimo v argumentu ukazov kot je npr. `\section`. Takšnim ukazom pravimo krhki ukazi. Krhka ukaza sta npr. `\footnote` in `\phantom`. Takšni krhki ukazi potrebujejo zaščito, da »ne razpadejo« in to dobimo z ukazom `\protect`, ki ga postavimo pred krhki ukaz.

Ukaz `\protect` deluje samo za ukaz, pred katerim stoji, ne velja pa niti za njegove argumente. V večini primerov preveč uporabljenih ukazov `\protect` ne more škoditi. Če v spodnjem primeru ne bi bilo ukaza `\protect`, bi \LaTeX pri prevajanju javil napako.

11. Tu smo predpostavili, da je čakalna vrsta za slike prazna.

```
\section{Delujem preišljeno  
  \protect\footnote{in varujem svoje opombe}}
```


Poglavje 3

Stavljenje matematičnih formul

Sedaj ste pripravljeni! V tem poglavju se bomo lotili najmočnejšega orožja v T_EXu: stavljenja matematičnih tekstov. Naj vas opozorim, da se v poglavju snovi le na grobo dotaknemo. Stvari, ki so omenjene, bodo zadoščale večini ljudi, če pa med njimi ne najdete rešitve za vaš problem stavljenja matematičnega teksta, ne obupajte. Obstaja velika verjetnost, da je vaš problem obdelan v paketu $\mathcal{A}\mathcal{M}\mathcal{S}$ -L_AT_EX.¹

3.1 Uvod

L_AT_EX ima poseben način za stavljenje matematičnih izrazov. Matematični tekst lahko znotraj odstavka vnesemo v t.i. *vrstičnem načinu* med $\backslash($ in $\backslash)$, med $\$$ in $\$$ ali med \backslashbegin{math} in \backslashend{math} . Vsi trije načini so ekvivalentni.

Vsota a^2 na kvadrat in b^2 na kvadrat je c^2 na kvadrat. Oziroma, če zapišemo bolj matematično: $\backslash(c^2=a^2+b^2)$.

Vsota a na kvadrat in b na kvadrat je c na kvadrat. Oziroma, če zapišemo bolj matematično: $c^2 = a^2 + b^2$.

\backslashTeX se izgovarja kot $\tau\epsilon\chi$.
 $\backslash\tau\epsilon\chi$ vode.
To prihaja iz mojega \heartsuit .

$\tau\epsilon\chi$ se izgovarja kot $\tau\epsilon\chi$.
100 m³ vode.
To prihaja iz mojega \heartsuit .

1. Ameriško matematično društvo (*American Mathematical Society*) je pripravilo močno nadgradnjo L_AT_EXa. Številni primeri v tem poglavju so bili narejeni s pomočjo te razširitve. V zadnjem času je vključena v vse distribucije T_EXa. Če jo vseeno pogrešate, jo lahko naložite na `macros/latex/required/amslatex`.

Večje matematične enačbe oziroma formule lahko ločimo od preostanka odstavka tako, da jih *prikažemo* namesto, da jih pišemo med tekstom. To naredimo tako, da izraz pišemo v t.i. *prikaznem načinu* med `\[` in `\]`, med `$$` in `$$` ali pa med `\begin{displaymath}` in `\end{displaymath}`. Vsi trije načini so ekvivalentni.

Vsota a na kvadrat in b na kvadrat je c na kvadrat. Oziroma, če zapišemo bolj matematično:

```
\begin{displaymath}
c^2=a^2+b^2
\end{displaymath}
```

Pišemo lahko tudi $a+b=c$.

Vsota a na kvadrat in b na kvadrat je c na kvadrat. Oziroma, če zapišemo bolj matematično:

$$c^2 = a^2 + b^2$$

Pišemo lahko tudi

$$a + b = c.$$

Če želimo, da L^AT_EX oštevilči enačbe, uporabimo okolje `equation`. Potem lahko z `\label` poimenujemo enačbo in se nato sklicujemo nanjo drugje v dokumentu z uporabo ukaza `\ref` ali `\eqref` iz paketa `amsmath`, ki avtomatično da oznako med oklepaje:

```
\begin{equation} \label{eq:eps}
\epsilon > 0
\end{equation}
Iz (\ref{eq:eps}), sledi
\ldots}S pomočjo \eqref{eq:eps}
lahko naredimo isto.
```

$$\epsilon > 0 \quad (3.1)$$

Iz (3.1), sledi ... S pomočjo (3.1) lahko naredimo isto.

Bodite pozorni na to, da se izraz stavi drugače v prikaznem kot v vrstičnem načinu:

```
$$\lim_{n \to \infty}
\sum_{k=1}^n \frac{1}{k^2}
= \frac{\pi^2}{6}$$
```

$$\lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6}$$

```
\begin{displaymath}
\lim_{n \to \infty}
\sum_{k=1}^n \frac{1}{k^2}
= \frac{\pi^2}{6}
\end{displaymath}
```

$$\lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6}$$

Obstajajo razlike med *matematičnim načinom* in *tekstovnim načinom*. V matematičnem načinu tako npr. velja:

1. Večina presledkov in prelomov vrstic nima nobenega pomena. Vsi presledki sledijo ali iz matematičnih izrazov ali pa je zanje potrebno uporabiti posebne ukaze, kot so npr. `\,`, `\quad` ali `\qquad`.

2. Prazne vrstice niso dovoljene. Formula ne more biti sestavljena iz več odstavkov.
3. Vsaka črka se obravnava kot ime spremenljivke in se tako tudi stavi v ustrezni pisavi. Če želimo v matematičnem načinu v formuli pisati normalen tekst (normalna pokončna pisava z normalnimi presledki), potem je potrebno tak tekst vnesti s pomočjo ukaza `\textrm{...}` (poglejte tudi razdelek 3.7 na strani 53).
4. Če želimo kot spremenljivke uporabljati šumnike oz. druge črke z akcenti, potem moramo v matematičnem načinu uporabljati matematične akcente. Namesto `\v{c}` moramo tako pisati `\check{c}`.

```
\begin{equation}
\forall x \in \mathbf{R}:
\quad x^2 \geq 0
\end{equation}
```

$$\forall x \in \mathbf{R} : \quad x^2 \geq 0 \quad (3.2)$$

```
\begin{equation}
x^2 \geq 0 \quad \text{za vsak } x \in \mathbf{R}
\end{equation}
```

$$x^2 \geq 0 \quad \text{za vsak } x \in \mathbf{R} \quad (3.3)$$

```
\begin{equation}
a+b+c+\check{c}+d=e
\end{equation}
```

$$a + b + c + \check{c} + d = e \quad (3.4)$$

Matematiki so lahko zelo sitni glede simbolov, ki naj se jih uporablja. Lep primer sta simbola za realna in kompleksna števila. Dogovor je, da naj se uporabljajo krepki simboli dobljeni z ukazom `\mathbb` iz paketa `amsmath` ali `amssymb`. Predzadnji primer se tako spremeni v

```
\begin{displaymath}
x^2 \geq 0 \quad \text{za vsak } x \in \mathbb{R}
\end{displaymath}
```

$$x^2 \geq 0 \quad \text{za vsak } x \in \mathbb{R}$$

3.2 Združevanje v matematičnem načinu

Ukazi v \LaTeX u, ki potrebujejo argument, pričakujejo, da je argument med zavirami oklepaji, sicer pa kot argument vzamejo prvo naslednjo črko. To velja tudi v matematičnem načinu, kjer večina ukazov deluje le na naslednjem znaku. Če želimo, da ukaz deluje na več znakih skupaj, jih združimo tako, da jih damo med zavirane oklepaje: `{...}`.

```
\begin{equation}
a^{x+y} \neq a^x a^y
\end{equation}
```

$$a^x + y \neq a^{x+y} \quad (3.5)$$

3.3 Osnovni gradniki matematičnih formul

V tem razdelku bomo opisali najpomembnejše ukaze za stavljenje matematičnih izrazov. Za podroben seznam vseh ukazov si poglejte razdelek 3.10 na strani 57.

Male grške črke vnašamo kot `\alpha`, `\beta`, `\gamma`, ..., velike grške črke pa kot `\Gamma`, `\Delta`, ...²

```
 $\lambda, \xi, \pi, \mu, \Phi, \Omega$
```

 $\lambda, \xi, \pi, \mu, \Phi, \Omega$

Potence in indekse vnašamo s pomočjo znakov `^` in `_`.

```
 $a_{1}$ \quad $x^{2}$ \quad $e^{-\alpha t}$ \quad $a^{3}_{ij}$ \quad $e^{x^2} \neq e^{x^2}$
```

 $a_1 \quad x^2 \quad e^{-\alpha t} \quad a_{ij}^3$
 $e^{x^2} \neq e^{x^2}$

Kvadratni koren vnesemo kot `\sqrt{x}`, n -ti koren pa z ukazom `\sqrt[n]{x}`. Velikost znaka za koren avtomatično določi L^AT_EX. Če potrebujemo le znak za koren, uporabimo `\surd`.

```
 $\sqrt{x}$ \quad $\sqrt{x^2 + \sqrt{y}}$ \quad $\sqrt[3]{2}$ \quad $\surd[x^2 + y^2]$
```

 $\sqrt{x} \quad \sqrt{x^2 + \sqrt{y}} \quad \sqrt[3]{2}$
 $\sqrt[x^2 + y^2]$

Ukaza `\overline{}` in `\underline{}` naredita **vodoravno črto** nad oziroma pod izrazom.

```
 $\overline{m+n}$ in $\underline{a+b}$
```

 $\overline{m+n} \text{ in } \underline{a+b}$

Ukaza `\overbrace{}` in `\underbrace{}` naredita **vodoravni zaviti oklepaj**, ki združuje elemente izraza, nad oziroma pod izrazom. Zaviti oklepaj lahko po želji dodatno opremimo z indeksom.

2. Pri velikih črkah imamo le tiste, ki se razlikujejo od latinskih, zato v L^AT_EX 2_ε ni velikega Alpha, saj je to kar veliki A. Stvari se bodo spremenile, ko bo pripravljeno novo kodiranje matematičnih znakov.

```


$$\underbrace{a+b+\cdots+z}_{26}$$

in

$$\overbrace{1+1+\cdots+1}^{17}$$


```

$$\underbrace{a+b+\cdots+z}_{26} \text{ in } \overbrace{1+1+\cdots+1}^{17}$$

Če želimo znakom dodati matematične akcente, kot sta npr. puščica za vektorje in tilda, potem uporabimo ukaze, ki so podani v tabeli 3.1 na strani 57. Strešice in tilde, ki se raztezajo čez več znakov, dobimo z ukazoma `\widehat` in `\widetilde`. Znak ' uporabljamo za odvod oz. za spremenljivke »s črtico«.

```


$$y=x^2 \quad y'=2x \quad y''=2$$


```

$$y = x^2 \quad y' = 2x \quad y'' = 2$$

Vektorje običajno pišemo tako, da nad izrazom pišemo puščico. Za to imamo na voljo ukaz `\vec`. Ukaza `\overrightarrow` in `\overleftarrow` prideta v poštev za daljše izraze, kot je npr. vektor, ki gre od točke *A* do *B*.

```


$$\vec{a} \quad \overrightarrow{AB}$$


```

$$\vec{a} \quad \overrightarrow{AB}$$

Pri množenju ponavadi ne pišemo pike med izrazoma. V nekaterih primerih pa je to vseeno potrebno, da se bralec lažje znajde. Tedaj uporabimo ukaz `\cdot`

```


$$v = \sigma_1 \cdot \sigma_2 \tau_1 \cdot \tau_2$$


```

$$v = \sigma_1 \cdot \sigma_2 \tau_1 \cdot \tau_2$$

Imena funkcij, kot so logaritem, sinus, ..., ponavadi pišemo v pokončni pisavi in ne poševno kot spremenljivke. Zato ima \LaTeX naslednje ukaze za večino najpomembnejših matematičnih funkcij:

<code>\arccos</code>	<code>\cos</code>	<code>\csc</code>	<code>\exp</code>	<code>\ker</code>	<code>\limsup</code>	<code>\min</code>
<code>\arcsin</code>	<code>\cosh</code>	<code>\deg</code>	<code>\gcd</code>	<code>\lg</code>	<code>\ln</code>	<code>\Pr</code>
<code>\arctan</code>	<code>\cot</code>	<code>\det</code>	<code>\hom</code>	<code>\lim</code>	<code>\log</code>	<code>\sec</code>
<code>\arg</code>	<code>\coth</code>	<code>\dim</code>	<code>\inf</code>	<code>\liminf</code>	<code>\max</code>	<code>\sin</code>
<code>\sinh</code>	<code>\sup</code>	<code>\tan</code>	<code>\tanh</code>			

```


$$\lim_{x \rightarrow 0} \frac{\sin x}{x} = 1$$


```

$$\lim_{x \rightarrow 0} \frac{\sin x}{x} = 1$$

Za modulsko funkcijo imamo dva ukaza : `\bmod` za binarni operator » $a \bmod b$ « in `\pmod` za izraze kot npr. » $x \equiv a \pmod{b}$ «.

Ulomke pišemo z ukazom `\frac{...}{...}`. Pogosto v tekstovnem načinu za ulomke uporabimo kar obliko $1/2$ namesto $\frac{1}{2}$, saj zglada lepše.

```
$\frac{1}{2}$~ura
\begin{displaymath}
\frac{x^2}{k+1} \quad
x^{\frac{2}{k+1}} \quad
x^{1/2}
\end{displaymath}
```

$1\frac{1}{2}$ ura

$$\frac{x^2}{k+1} \quad x^{\frac{2}{k+1}} \quad x^{1/2}$$

Za binomske koeficiente in podobne strukture imamo na voljo ukaza `\binom{...}{...}` in `\atop`. Drugi ukaz vrne podoben rezultat kot prvi, le brez oklepajev.³

```
\begin{displaymath}
\binom{n}{k} \quad
x \atop y+2
\end{displaymath}
```

$$\binom{n}{k} \quad x \atop y+2$$

Pri binarnih relacijah je pomembno znati postaviti izraze drug na drugega. Ukaz `\stackrel{!}{=}` postavi podani prvi argument v velikosti enaki velikosti potenc na drugi argument, ki je v normalni velikosti.

```
\begin{displaymath}
\int f_N(x) \stackrel{!}{=} 1
\end{displaymath}
```

$$\int f_N(x) \stackrel{!}{=} 1$$

Znak za **integral** dobimo z `\int`, za **vsoto** s `\sum` in za **produkt** s `\prod`. Zgornjo in spodnjo mejo podamo z `^` in `_`, tako kot potence in indekse.⁴

```
\begin{displaymath}
\sum_{i=1}^n \quad
\int_0^{\frac{\pi}{2}} \quad
\prod_{\epsilon}
\end{displaymath}
```

$$\sum_{i=1}^n \quad \int_0^{\frac{\pi}{2}} \quad \prod_{\epsilon}$$

Za še boljši nadzor nad postavitvijo indeksov v kompleksnih izrazih inamo v paketu `amsmath` na voljo še dodatni orodji: ukaz `\substack` in okolje `subarray`:

3. Uporaba teh starih izrazov je prepovedana v paketu `amsmath`, kjer sta nadomeščena z `\binom` in `\genfrac`.

4. `AMS-LATEX` pozna tudi zgornje in spodnje indekse v več vrsticah.

```

\begin{displaymath}
\sum_{\substack{0 < i < n \\ 1 < j < m}}
P(i, j) =
\sum_{\begin{subarray}{l} i \in I \\ 1 < j < m \end{subarray}}
Q(i, j)
\end{displaymath}

```

$$\sum_{\substack{0 < i < n \\ 1 < j < m}} P(i, j) = \sum_{\substack{i \in I \\ 1 < j < m}} Q(i, j)$$

Za **oklepaje** in druge razmejitvene simbole imamo v \TeX u vse vrste simbolov (npr. $\langle \parallel \Downarrow$). Okrogle in oglete oklepaje dobimo z ustreznimi tipkami, zavite z $\{\}$, za ostale pa uporabimo posebne ukaze (npr. \updownarrow). Za seznam vseh možnosti pogledajte tabelo 3.7 na strani 59.

```

\begin{displaymath}
\{a, b, c\} \neq \{a, b, c\}
\end{displaymath}

```

$$a, b, c \neq \{a, b, c\}$$

Če pred začetni oklepaj postavimo ukaz \left , pred zadnjega pa \right , bo \TeX avtomatično določil velikost oklepajev. Vedno mora vsak \left imeti tudi svoj \right , velikost pa bo pravilna le, če bosta oba uporabljena v isti vrstici. Če želimo imeti oklepaj le na eni strani, potem na drugi strani namesto znaka za oklepaj vstavimo piko, ki pomeni nevidni oklepaj. Če npr. želimo oklepaj le na levi, potem na desni uporabimo $\right.$.

```

\begin{displaymath}
1 + \left( \frac{1}{1-x^2} \right)^3
\end{displaymath}

```


$$1 + \left(\frac{1}{1-x^2} \right)^3$$

V nekaterih primerih moramo sami ročno določiti velikost oklepajev, kar lahko naredimo z ukazi \big , \Big , \bigg in \Bigg , ki jih uporabimo pred večino oklepajev.⁵

```

\Big( (x+1)(x-1) \Big)^2 \\
\big(\Big(\bigg(\Big(\quad \\
\big)\Big)\bigg)\Big)\quad \\
\big|\Big|\bigg|\Big|

```

$$\left((x+1)(x-1) \right)^2$$


Za vnos **treh pik** v formulo imamo na voljo več ukazov. Ukaz \ldots izpiše pike na dnu, \cdots da pike na sredo, poleg tega pa imamo še ukaza \vdots za navpične pike in \ddots za diagonalne pike. Več primerov lahko najdete v razdelku 3.5.

5. Ti ukazi ne delujejo pravilno, če spreminjamo velikost pisave ali če uporabljamo velikost 11pt ali 12pt. To se da popraviti z uporabo paketa `exscale` ali `amsmath`.

```
\begin{displaymath}
x_{1},\ldots,x_{n} \quad \quad
x_{1}+\cdots+x_{n}
\end{displaymath}
```

$$x_1, \dots, x_n \quad x_1 + \dots + x_n$$

3.4 Presledki v matematičnem načinu

Če presledki v enačbah, ki jih izbere T_EX niso zadovoljivi, jih lahko popravimo s pomočjo posebnih ukazov. Tako imamo na voljo nekaj ukazov za majhne presledke: `\`, za $\frac{3}{18}$ quad (\mathbb{U}), `\:` za $\frac{4}{18}$ quad (\mathbb{U}) in `\;` za $\frac{5}{18}$ quad (\mathbb{U}). Standardni ukaz za presledek `_` naredi srednje velik presledek, ukaza `\quad` (\mathbb{U}) in `\qquad` (\mathbb{U}) pa velika presledka. Velikost `\quad` ustreza širini znaka 'M' v trenutni pisavi. Ukaz `\!` naredi negativni presledek s širino $-\frac{3}{18}$ quad (\mathbb{U}).

```
\newcommand{\ud}{\mathrm{d}}
\begin{displaymath}
\int\!\!\!\int_{D} g(x,y)
\quad \quad \quad
\int\int_{D} g(x,y)\ud x \ud y
\end{displaymath}
namesto
\begin{displaymath}
\int\int_{D} g(x,y)\ud x \ud y
\end{displaymath}
```

namesto

$$\iint_D g(x,y) dx dy$$

$$\int \int_D g(x,y) dx dy$$

Bodite pozorni na to, da je znak za diferencial 'd' zapisan v pokončni pisavi.

V $\mathcal{A}\mathcal{M}\mathcal{S}\text{-L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ u imamo na voljo dodatne ukaze za fino nastavitvev presledkov med večkratnimi znaki za integriranje, to so ukazi `\iint`, `\iiint`, `\iiiiint` in `\idotsint`. Če naložimo paket `amsmath`, lahko zgornji primer sestavimo tudi na naslednji način:

```
\newcommand{\ud}{\mathrm{d}}
\begin{displaymath}
\iint_{D} \quad \quad \quad
\iint_{D} \ud x \ud y
\end{displaymath}
```

$$\iint_D dx dy$$

Za več podrobnosti pogledjte dokument `testmath.tex` (vsebovan je v paketu $\mathcal{A}\mathcal{M}\mathcal{S}\text{-L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$) ali poglavje 8 v »The LaTeX Companion«.

3.5 Navpično poravnavanje

Za sestavljanje **matematičnih razpredelnic** uporabljamo okolje `array`. Deluje podobno kot okolje `tabular`. Za prelom vrstice uporabljamo ukaz `\\`.


```
\begin{displaymath}
\mathbf{X} =
\left( \begin{array}{ccc}
x_{11} & x_{12} & \dots \\
x_{21} & x_{22} & \dots \\
\vdots & \vdots & \ddots
\end{array} \right)
\end{displaymath}
```

$$\mathbf{X} = \begin{pmatrix} x_{11} & x_{12} & \dots \\ x_{21} & x_{22} & \dots \\ \vdots & \vdots & \ddots \end{pmatrix}$$

Okolje `array` lahko uporabljamo tudi za sestavljanje izrazov, ki imajo na eni strani en velik oklepaj, če na drugi strani uporabimo prazni oklepaj, kot npr. ‘`\right.`’.

```
\begin{displaymath}
y = \left\{ \begin{array}{ll}
a & \text{če } d > c \\
b+x & \text{zjutraj} \\
l & \text{čez cel dan}
\end{array} \right.
\end{displaymath}
```

$$y = \begin{cases} a & \text{če } d > c \\ b + x & \text{zjutraj} \\ l & \text{čez cel dan} \end{cases}$$

Kot v okolju `tabular` lahko tudi v okolju `array` rišemo navpične in vododravne črte, ki npr. ločujejo elemente matrice:

```
\begin{displaymath}
\left( \begin{array}{c|c}
1 & 2 \\ \hline
3 & 4
\end{array} \right)
\end{displaymath}
```

$$\left(\begin{array}{c|c} 1 & 2 \\ \hline 3 & 4 \end{array} \right)$$

Za formule, ki se raztezajo čez več vrstic oz. za sisteme enačb uporabljamo okolji `eqnarray` in `eqnarray*` namesto `equation`. V `eqnarray` se vsaka vrstica avtomatično oštevilči, pri `eqnarray*` pa se nobena vrstica ne oštevilči.

Okolji `eqnarray` in `eqnarray*` delujeta kot razpredelnica s tremi stolpci oblike `{rc1}`, kjer se srednji stolpec uporablja za enačaj, neenačaj, ali pa katerikoli drugi znak, po katerem želimo poravnati vrstice. Ukaz `\` pomeni prehod v novo vrstico.

```
\begin{eqnarray}
f(x) & = & \cos x & \\
f'(x) & = & -\sin x & \\
\int_0^x f(y)dy & = & \sin x & \\
\end{eqnarray}
```

$$f(x) = \cos x \quad (3.6)$$

$$f'(x) = -\sin x \quad (3.7)$$

$$\int_0^x f(y)dy = \sin x \quad (3.8)$$

Opazimo lahko, da je presledek tako na levi kot tudi na desni strani enačaja precej velik. To lahko zmanjšamo s `\setlength\arraycolsep{2pt}`, kot je to prikazano v naslednjem zgledu.

Dolge enačbe se avtomatično ne delijo lepo. Avtor mora sam povedati, kje naj se začne nova vrstica in kolikšen naj bo začetni zamik. V ta namen se najpogosteje uporabljata naslednji dve metodi.

```
{\setlength\arraycolsep{2pt}
\begin{eqnarray}
\sin x & = & x - \frac{x^3}{3!} + \frac{x^5}{5!} - \\
& & \frac{x^7}{7!} + \dots
\end{eqnarray}}
```

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots \quad (3.9)$$

```
\begin{eqnarray}
\lefteqn{ \cos x = 1 } \\
& - \frac{x^2}{2!} + \\
& \frac{x^4}{4!} - \\
& \frac{x^6}{6!} + \dots
\end{eqnarray}}
```

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots \quad (3.10)$$

Ukaz `\nonumber` pove \LaTeX u, da naj ne oštevilči enačbe. Argument ukaza `\lefteqn` \LaTeX izpiše, vendar ga obravnava, kot da ima širino 0. Prva vrstica v zadnjem primeru je tako sestavljena le iz levega stolpca, ker pa ima za \LaTeX širino 0, se naslednja vrstica začne že pred koncem zgornje vrstice.

Če z navedenimi metodami še vedno ne uspemo pravilno navpično poravnati enačb, potem v paketu `amsmath` obstajajo še močnejši ukazi (poglejte okolji `split` in `align`).

3.6 Fantomski objekti

Fantomskih objektov ne vidimo, kljub temu pa zavzemajo prostor. Z njimi lahko v \LaTeX u izvedemo zanimive trike.

Ko navpično poravnavamo tekst, ki vsebuje `^` in `_`, je \LaTeX dostikrat celo preveč uslužen. S pomočjo ukaza `\phantom` lahko rezerviramo prostor za znake, ki se v končnem rezultatu ne pokažejo. Najbolje, da pogledamo kar naslednja zgleda.

```
\begin{displaymath}
{}^{\phantom{1}6}\text{C}
\quad \text{ali} \quad
{}^{\phantom{1}6}\text{C}
\end{displaymath}
```

$${}^{6}\text{C} \quad \text{ali} \quad {}^{6}\text{C}$$

```
\begin{displaymath}
\Gamma_{ij}^k
\quad \text{ali} \quad
\Gamma_{ij}^k
\end{displaymath}
```

$$\Gamma_{ij}^k \quad \text{ali} \quad \Gamma_{ij}^k$$

3.7 Velikost pisave v matematičnem načinu

V matematičnem načinu $\text{T}_{\text{E}}\text{X}$ določi velikost pisave glede na kontekst. Potence in indeksi se npr. izpišejo z manjšo velikostjo. Če želite del enačbe izpisati pokončno, potem ne uporabite ukaza `\textterm`, saj v tem primeru spreminjanje velikosti pisave ne deluje, saj `\textterm` začasno prestavi v tekstovni način. Pravilno je uporabiti `\mathrm`, saj se tu velikost pisave avtomatično prilagaja. Paziti pa je potrebno, da `\mathrm` deluje dobro le na kratkih izrazih. Presledki in znaki z akcenti ne delujejo.⁶

```
\begin{equation}
2^{\text{nd}} \quad \text{ali} \quad
2^{\mathrm{nd}}
\end{equation}
```

$$2^{\text{nd}} \quad 2^{\mathrm{nd}} \quad (3.11)$$

Kljub temu je v $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ u včasih potrebno ročno določiti pravilno velikost pisave. V matematičnem načinu imamo za to na voljo naslednje štiri ukaze:

```
\displaystyle (123), \textstyle (123), \scriptstyle (123) in
\scriptscriptstyle (123).
```

Spreminjanje velikosti vpliva tudi na to, kako se izpisujejo meje.

```
\begin{displaymath}
\mathop{\mathrm{corr}}(X, Y) =
\frac{\displaystyle
\sum_{i=1}^n (x_i - \overline{x})
(y_i - \overline{y})}
{\displaystyle \biggl[
\sum_{i=1}^n (x_i - \overline{x})^2
\sum_{i=1}^n (y_i - \overline{y})^2
\biggr]^{1/2}}
\end{displaymath}
```

$$\text{corr}(X, Y) = \frac{\sum_{i=1}^n (x_i - \overline{x})(y_i - \overline{y})}{\left[\sum_{i=1}^n (x_i - \overline{x})^2 \sum_{i=1}^n (y_i - \overline{y})^2 \right]^{1/2}}$$

⁶ Če vključimo paket $\mathcal{A}\mathcal{M}\mathcal{S}\text{-L}_{\text{A}}\text{T}_{\text{E}}\text{X}$, potem ukaz `\textterm` deluje s prilagajanjem velikosti pisave.

To je eden izmed primerov, kjer potrebujemo večji oklepaj od tistega, ki ga dobimo iz `\left[\right]`.

3.8 Izreki, trditve, ...

Ko pišemo matematični tekst, potrebujemo način za stavljenje lem, definicij, izrekov, aksiomov in podobnih struktur. \LaTeX podpira to z ukazom

```
\newtheorem{ime}[stevec]{naslov}[section]
```

Argument *ime* je kratka ključna beseda, s katero povemo \LaTeX u za kakšno matematično trditev gre. V argumentu *naslov* navedemo dejansko ime trditve, ki se izpiše v prevedenem dokumentu.

Argumenti v oglatih oklepajih so neobvezni. Z njimi lahko določimo način oštevilčenja trditev. Argument *stevec* vsebuje ime trditve, po kateri naj se številči trditev. Izberemo lahko številčenje po že predhodno definirani trditvi ali pa vnesemo kar isto ime kot v parametru *ime*. Nove trditve se potem številčijo v istem zaporedju. Argument *section* je ime logične strukture znotraj katere številčimo trditve.

Če damo ukaz `\newtheorem` v preambulo dokumenta, lahko potem znotraj dokumenta uporabljamo okolje `\begin{ime} ... \end{ime}`

```
\begin{ime}[tekst]
To je moj zanimiv izrek.
\end{ime}
```

V paketu `amsthm` je na voljo ukaz `\newtheoremstyle{style}`, s katerim lahko definiramo vrsto matematične trditve z izbiro ene izmed treh vnaprej pripravljenih oblik: `definition` (krepak naslov, pokončna pisava), `plain` (krepak naslov, poševna pisava) ali `remark` (poševen naslov, pokončna pisava).

Dovolj teorije. Naslednji zgledi bodo razjasnili vse dvome in dokončno dokazali, da je okolje `\newtheorem` prezapleteno, da bi ga lahko razumeli.

Najprej definiramo nove matematične izjave:

```
\theoremstyle{definition} \newtheorem{izrek}{Izrek}
\theoremstyle{plain}      \newtheorem{posl}[izrek]{Posledica}
\theoremstyle{remark}     \newtheorem*{opomba}{Opomba}
```

```

\begin{izrek}[Pitagora]
\label{izrek:Pit}
V pravokotnem trikotniku velja
 $c^2=a^2+b^2$ .
\end{izrek}
\begin{posl}
Če velja  $c^2 \neq a^2+b^2$ , potem
trikotnik ni pravokoten
(poglej izrek~\ref{izrek:Pit}).
\end{posl}
\begin{opomba}To je bil moj
najnovejši izrek o \ldots
\end{opomba}

```

Izrek 1 (Pitagora). V pravokotnem trikotniku velja $c^2 = a^2 + b^2$.

Posledica 2. Če velja $c^2 \neq a^2 + b^2$, potem trikotnik ni pravokoten (poglej izrek 1).

Opomba. To je bil moj najnovejši izrek o ...

Trditev »Posledica« uporablja isti števec kot trditev »Izrek«, zato dobi številko iz istega zaporedja. Neobvezni argument v oglatih oklepajih uporabimo za to, da določimo naslov trditve ali kaj podobnega (npr. avtorja in referenco).

```

\flushleft
\newtheorem{mur}{Murphy}[section]
\begin{mur}
Če se da nekaj narediti na več
načinov in eden izmed njih vodi
v katastrofo, potem bo nekdo
izbral prav ta način.
\end{mur}

```

Murphy 3.8.1. Če se da nekaj narediti na več načinov in eden izmed njih vodi v katastrofo, potem bo nekdo izbral prav ta način.

Trditev »Murphy« dobi številko, ki je vezana na številko trenutnega razdelka. Namesto tega lahko uporabimo tudi kakšno drugo logično enoto, npr. poglavje ali podrazdelek.

V paketu `amsthm` je na voljo tudi okolje `proof` za dokaze.

```

\begin{proof}
Očitno, uporabimo
 $E=mc^2$ .
\end{proof}

```

Dokaz. Očitno, uporabimo

$$E = mc^2.$$

□

Z ukazom `\qedhere` lahko premaknemo znak za konec dokaza v situacijah, ko bi sicer stal na koncu prazne vrstice.

```

\begin{proof}
Očitno, uporabimo
 $E=mc^2$ . \qedhere
\end{proof}

```

Dokaz. Očitno, uporabimo

$$E = mc^2.$$

□

3.9 Krepki simboli

V \LaTeX u ni najbolj enostavno dobiti krepkih simbolov; to je najbrž namerno zaradi tega, ker jih začetniki preveč uporabljajo. Ukaz za spremembo pisave `\mathbf` sicer naredi krepke črke, toda to so pokončne črke, matematični simboli pa so ponavadi poševni. Obstaja ukaz `\boldmath`, toda tega *lahko uporabljamo le zunaj matematičnega načina*. Deluje pa tudi za simbole.

```
\begin{displaymath}
\mu, M \quad \mathbf{M} \quad \mu, M
\mbox{\boldmath $\mu, M$}
\end{displaymath}
```



$$\mu, M \quad \mathbf{M} \quad \mu, M$$

V zgornjem primeru je krepka tudi vejica, kar ni ravno to, kar bi radi.

Paket `amsbsy` (ki se avtomatično vključi z `amsmath`) kakor tudi paket `bm` olajšata zadevo z ukazom `\boldsymbol`.

```
\begin{displaymath}
\mu, M \quad \boldsymbol{\mu}, \boldsymbol{M}
\end{displaymath}
```



$$\mu, M \quad \boldsymbol{\mu}, \boldsymbol{M}$$

3.10 Seznam matematičnih simbolov

Naslednje tabele prikazujejo vse simbole, ki so normalno na voljo v *matematičnem načinu*.

Za uporabo simbolov iz tabel 3.11–3.15,⁷ morate v preambuli naložiti paket `amssymb`, na vašem sistemu pa morajo biti nameščene AMS pisave. Če tega nimate, pogledjte na `macros/latex/required/amslatex`. Še popolnejši seznam vseh simbolov lahko najdete na `info/symbols/comprehensive`.

Tabela 3.1: Matematični akcenti.

\hat{a}	<code>\hat{a}</code>	\check{a}	<code>\check{a}</code>	\tilde{a}	<code>\tilde{a}</code>
\grave{a}	<code>\grave{a}</code>	\acute{a}	<code>\dot{a}</code>	\ddot{a}	<code>\ddot{a}</code>
\bar{a}	<code>\bar{a}</code>	\vec{a}	<code>\vec{a}</code>	\widehat{A}	<code>\widehat{A}</code>
\acute{a}	<code>\acute{a}</code>	\breve{a}	<code>\breve{a}</code>	\widetilde{A}	<code>\widetilde{A}</code>

Tabela 3.2: Grške črke.

α	<code>\alpha</code>	θ	<code>\theta</code>	o	<code>o</code>	υ	<code>\upsilon</code>
β	<code>\beta</code>	ϑ	<code>\vartheta</code>	π	<code>\pi</code>	ϕ	<code>\phi</code>
γ	<code>\gamma</code>	ι	<code>\iota</code>	ϖ	<code>\varpi</code>	φ	<code>\varphi</code>
δ	<code>\delta</code>	κ	<code>\kappa</code>	ρ	<code>\rho</code>	χ	<code>\chi</code>
ϵ	<code>\epsilon</code>	λ	<code>\lambda</code>	ϱ	<code>\varrho</code>	ψ	<code>\psi</code>
ε	<code>\varepsilon</code>	μ	<code>\mu</code>	σ	<code>\sigma</code>	ω	<code>\omega</code>
ζ	<code>\zeta</code>	ν	<code>\nu</code>	ς	<code>\varsigma</code>		
η	<code>\eta</code>	ξ	<code>\xi</code>	τ	<code>\tau</code>		
Γ	<code>\Gamma</code>	Λ	<code>\Lambda</code>	Σ	<code>\Sigma</code>	Ψ	<code>\Psi</code>
Δ	<code>\Delta</code>	Ξ	<code>\Xi</code>	Υ	<code>\Upsilon</code>	Ω	<code>\Omega</code>
Θ	<code>\Theta</code>	Π	<code>\Pi</code>	Φ	<code>\Phi</code>		

7. Te tabele so dobljene iz datoteke `symbols.tex`, ki jo je napisal David Carlisle, potem pa popravil še Josef Tkadlec.

Tabela 3.3: Binarne relacije.

Ustrezne negacije dobimo, če pred relacijo dodamo ukaz `\not`.

$<$	<code><</code>	$>$	<code>></code>	$=$	<code>=</code>
\leq	<code>\leq</code> or <code>\le</code>	\geq	<code>\geq</code> or <code>\ge</code>	\equiv	<code>\equiv</code>
\ll	<code>\ll</code>	\gg	<code>\gg</code>	\doteq	<code>\doteq</code>
\prec	<code>\prec</code>	\succ	<code>\succ</code>	\sim	<code>\sim</code>
\preceq	<code>\preceq</code>	\succeq	<code>\succeq</code>	\simeq	<code>\simeq</code>
\subset	<code>\subset</code>	\supset	<code>\supset</code>	\approx	<code>\approx</code>
\subseteq	<code>\subseteq</code>	\supseteq	<code>\supseteq</code>	\cong	<code>\cong</code>
\sqsubset	<code>\sqsubset</code> ^a	\sqsupset	<code>\sqsupset</code> ^a	\Join	<code>\Join</code> ^a
\sqsubseteq	<code>\sqsubseteq</code>	\sqsupseteq	<code>\sqsupseteq</code>	\bowtie	<code>\bowtie</code>
\in	<code>\in</code>	\ni	<code>\ni</code> , <code>\owns</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>
\smile	<code>\smile</code>	\frown	<code>\frown</code>	\asymp	<code>\asymp</code>
$:$	<code>:</code>	\notin	<code>\notin</code>	\neq	<code>\neq</code> or <code>\ne</code>

^aZa ta simbol potrebujete paket `latexsym`

Tabela 3.4: Binarne operacije

$+$	<code>+</code>	$-$	<code>-</code>	\triangleleft	<code>\triangleleft</code>
\pm	<code>\pm</code>	\mp	<code>\mp</code>	\triangleleft	<code>\triangleright</code>
\cdot	<code>\cdot</code>	\div	<code>\div</code>	\star	<code>\star</code>
\times	<code>\times</code>	\setminus	<code>\setminus</code>	\ast	<code>\ast</code>
\cup	<code>\cup</code>	\cap	<code>\cap</code>	\circ	<code>\circ</code>
\sqcup	<code>\sqcup</code>	\sqcap	<code>\sqcap</code>	\bullet	<code>\bullet</code>
\vee	<code>\vee</code> , <code>\lor</code>	\wedge	<code>\wedge</code> , <code>\land</code>	\diamond	<code>\diamond</code>
\oplus	<code>\oplus</code>	\ominus	<code>\ominus</code>	\uplus	<code>\uplus</code>
\odot	<code>\odot</code>	\oslash	<code>\oslash</code>	\amalg	<code>\amalg</code>
\otimes	<code>\otimes</code>	\bigcirc	<code>\bigcirc</code>	\dagger	<code>\dagger</code>
\triangleleft	<code>\bigtriangleup</code>	\triangledown	<code>\bigtriangledown</code>	\ddagger	<code>\ddagger</code>
\triangleleft	<code>\lhd</code> ^a	\triangleright	<code>\rhd</code> ^a	\wr	<code>\wr</code>
\triangleleft	<code>\unlhd</code> ^a	\triangleright	<code>\unrhd</code> ^a		

Tabela 3.5: Veliki operatorji.

Σ	<code>\sum</code>	\cup	<code>\bigcup</code>	\vee	<code>\bigvee</code>
\prod	<code>\prod</code>	\cap	<code>\bigcap</code>	\wedge	<code>\bigwedge</code>
\coprod	<code>\coprod</code>	\sqcup	<code>\bigsqcup</code>	\oplus	<code>\bigoplus</code>
\int	<code>\int</code>	\oint	<code>\oint</code>	\odot	<code>\bigodot</code>
\oplus	<code>\bigoplus</code>	\otimes	<code>\bigotimes</code>		

Tabela 3.6: Puščice.

\leftarrow	<code>\leftarrow</code> or <code>\gets</code>	\longleftarrow	<code>\longleftarrow</code>
\rightarrow	<code>\rightarrow</code> or <code>\to</code>	\longrightarrow	<code>\longrightarrow</code>
\leftrightarrow	<code>\leftrightarrow</code>	\longleftrightarrow	<code>\longleftrightarrow</code>
\Leftarrow	<code>\Leftarrow</code>	\Lleftarrow	<code>\Lleftarrow</code>
\Rightarrow	<code>\Rightarrow</code>	\Rrightarrow	<code>\Rrightarrow</code>
\Leftrightarrow	<code>\Leftrightarrow</code>	\Llongleftrightarrow	<code>\Llongleftrightarrow</code>
\mapsto	<code>\mapsto</code>	\longmapsto	<code>\longmapsto</code>
\hookrightarrow	<code>\hookrightarrow</code>	\hookrightarrow	<code>\hookrightarrow</code>
\lleftarrow	<code>\lleftarrow</code>	\rightharpoonup	<code>\rightharpoonup</code>
\lharpoonup	<code>\lharpoonup</code>	\rightharpoonup	<code>\rightharpoonup</code>
\leftharpoonup	<code>\leftharpoonup</code>	\rightharpoonup	<code>\rightharpoonup</code>
\rightleftharpoons	<code>\rightleftharpoons</code>	\iff	<code>\iff</code> (bigger spaces)
\uparrow	<code>\uparrow</code>	\downarrow	<code>\downarrow</code>
\Uparrow	<code>\Uparrow</code>	\Uparrow	<code>\Uparrow</code>
\Downarrow	<code>\Downarrow</code>	\Downarrow	<code>\Downarrow</code>
\nearrow	<code>\nearrow</code>	\searrow	<code>\searrow</code>
\swarrow	<code>\swarrow</code>	\nwarrow	<code>\nwarrow</code>
\leadsto	<code>\leadsto</code> ^a		

^aZa ta simbol potrebujete paket `latexsym`

Tabela 3.7: Oklepaji.

$($	<code>(</code>	$)$	<code>)</code>	\uparrow	<code>\uparrow</code>
$[$	<code>[</code> or <code>\lbrack</code>	$]$	<code>]</code> or <code>\rbrack</code>	\downarrow	<code>\downarrow</code>
$\{$	<code>\{</code> or <code>\lbrace</code>	$\}$	<code>\}</code> or <code>\rbrace</code>	\updownarrow	<code>\updownarrow</code>
\langle	<code>\langle</code>	\rangle	<code>\rangle</code>	$ $	<code> </code> or <code>\vert</code>
\lfloor	<code>\lfloor</code>	\rfloor	<code>\rfloor</code>	\lceil	<code>\lceil</code>
$/$	<code>/</code>	\backslash	<code>\backslash</code>	\Updownarrow	<code>\Updownarrow</code>
\Uparrow	<code>\Uparrow</code>	\Downarrow	<code>\Downarrow</code>	$\ $	<code>\ </code> or <code>\Vert</code>
\lceil	<code>\lceil</code>				

Tabela 3.8: Veliki oklepaji.

$($	<code>\lgroup</code>	$)$	<code>\rgroup</code>	\int	<code>\lmoustache</code>
$ $	<code>\arrowvert</code>	$\ $	<code>\Arrowvert</code>	$ $	<code>\bracevert</code>
$\}$	<code>\rmoustache</code>				

Tabela 3.9: Razni simboli.

\dots	<code>\dots</code>	\cdots	<code>\cdots</code>	\vdots	<code>\vdots</code>	\ddots	<code>\ddots</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>
\forall	<code>\forall</code>	\exists	<code>\exists</code>	\mho	<code>\mho</code> ^a	∂	<code>\partial</code>
$'$	<code>'</code>	$'$	<code>\prime</code>	\emptyset	<code>\emptyset</code>	∞	<code>\infty</code>
∇	<code>\nabla</code>	\triangle	<code>\triangle</code>	\square	<code>\Box</code> ^a	\diamond	<code>\Diamond</code> ^a
\perp	<code>\bot</code>	\top	<code>\top</code>	\sphericalangle	<code>\angle</code>	\surd	<code>\surd</code>
\diamond	<code>\diamondsuit</code>	\heartsuit	<code>\heartsuit</code>	\clubsuit	<code>\clubsuit</code>	\spadesuit	<code>\spadesuit</code>
\neg	<code>\neg</code> or <code>\lnot</code>	\flat	<code>\flat</code>	\natural	<code>\natural</code>	\sharp	<code>\sharp</code>

^aZa ta simbol potrebujete paket `latexsym`

Tabela 3.10: Nematematični simboli.

These symbols can also be used in text mode.

\dagger	<code>\dag</code>	\S	<code>\S</code>	\copyright	<code>\copyright</code>	\textregistered	<code>\textregistered</code>
\ddagger	<code>\ddag</code>	\P	<code>\P</code>	\pounds	<code>\pounds</code>	$\%$	<code>\%</code>

Tabela 3.11: AMS oklepaji.

\ulcorner	<code>\ulcorner</code>	\urcorner	<code>\urcorner</code>	\llcorner	<code>\llcorner</code>	\lrcorner	<code>\lrcorner</code>
$ $	<code>\lvert</code>	$ $	<code>\rvert</code>	$\ $	<code>\lVert</code>	$\ $	<code>\rVert</code>

Tabela 3.12: AMS grške in hebrejske črke.

\digamma	<code>\digamma</code>	\varkappa	<code>\varkappa</code>	\beth	<code>\beth</code>	\gimel	<code>\gimel</code>	\daleth	<code>\daleth</code>
------------	-----------------------	-------------	------------------------	---------	--------------------	----------	---------------------	-----------	----------------------

Tabela 3.13: AMS binarne relacije.

\lessdot	<code>\lessdot</code>	\gtrdot	<code>\gtrdot</code>	\doteqdot	<code>\doteqdot</code>
\leqslant	<code>\leqslant</code>	\geqslant	<code>\geqslant</code>	\risingdotseq	<code>\risingdotseq</code>
\eqslantless	<code>\eqslantless</code>	\eqslantgtr	<code>\eqslantgtr</code>	\fallingdotseq	<code>\fallingdotseq</code>
\leqq	<code>\leqq</code>	\geqq	<code>\geqq</code>	\eqcirc	<code>\eqcirc</code>
\lll or \llless	<code>\lll</code> or <code>\llless</code>	\ggg	<code>\ggg</code>	\circeq	<code>\circeq</code>
\lesssim	<code>\lesssim</code>	\gtrsim	<code>\gtrsim</code>	\triangleq	<code>\triangleq</code>
\lessapprox	<code>\lessapprox</code>	\gtrapprox	<code>\gtrapprox</code>	\bumpeq	<code>\bumpeq</code>
\lessgtr	<code>\lessgtr</code>	\gtrless	<code>\gtrless</code>	\Bumpeq	<code>\Bumpeq</code>
\lesseqgtr	<code>\lesseqgtr</code>	\gtreqless	<code>\gtreqless</code>	\thicksim	<code>\thicksim</code>
\lesseqqgtr	<code>\lesseqqgtr</code>	\gtreqqless	<code>\gtreqqless</code>	\thickapprox	<code>\thickapprox</code>
\preccurlyeq	<code>\preccurlyeq</code>	\succcurlyeq	<code>\succcurlyeq</code>	\approxeq	<code>\approxeq</code>
\curlyeqprec	<code>\curlyeqprec</code>	\curlyeqsucc	<code>\curlyeqsucc</code>	\backsim	<code>\backsim</code>
\precsim	<code>\precsim</code>	\succsim	<code>\succsim</code>	\backsimeq	<code>\backsimeq</code>
\precapprox	<code>\precapprox</code>	\succapprox	<code>\succapprox</code>	\vDash	<code>\vDash</code>
\subseteqq	<code>\subseteqq</code>	\supseteqq	<code>\supseteqq</code>	\Vdash	<code>\Vdash</code>
\shortparallel	<code>\shortparallel</code>	\Supset	<code>\Supset</code>	\Vvdash	<code>\Vvdash</code>
\blacktriangleleft	<code>\blacktriangleleft</code>	\sqsupset	<code>\sqsupset</code>	\backepsilon	<code>\backepsilon</code>
\vartriangleright	<code>\vartriangleright</code>	\because	<code>\because</code>	\varpropto	<code>\varpropto</code>
\blacktriangleright	<code>\blacktriangleright</code>	\Subset	<code>\Subset</code>	\between	<code>\between</code>
\trianglerighteq	<code>\trianglerighteq</code>	\smallfrown	<code>\smallfrown</code>	\pitchfork	<code>\pitchfork</code>
\vartriangleleft	<code>\vartriangleleft</code>	\shortmid	<code>\shortmid</code>	\smallsmile	<code>\smallsmile</code>
\trianglelefteq	<code>\trianglelefteq</code>	\therefore	<code>\therefore</code>	\sqsubset	<code>\sqsubset</code>

Tabela 3.14: AMS puščice.

\dashleftarrow	<code>\dashleftarrow</code>	\dashrightarrow	<code>\dashrightarrow</code>
\Lleftarrow	<code>\Lleftarrow</code>	\Rrightarrow	<code>\Rrightarrow</code>
\twoheadleftarrow	<code>\twoheadleftarrow</code>	\twoheadrightarrow	<code>\twoheadrightarrow</code>
\leftarrowtail	<code>\leftarrowtail</code>	\rightarrowtail	<code>\rightarrowtail</code>
\leftrightharpoons	<code>\leftrightharpoons</code>	\rightleftharpoons	<code>\rightleftharpoons</code>
\Lsh	<code>\Lsh</code>	\Rsh	<code>\Rsh</code>
\looparrowleft	<code>\looparrowleft</code>	\looparrowright	<code>\looparrowright</code>
\curvearrowleft	<code>\curvearrowleft</code>	\curvearrowright	<code>\curvearrowright</code>
\circlearrowleft	<code>\circlearrowleft</code>	\circlearrowright	<code>\circlearrowright</code>
\multimap	<code>\multimap</code>	\Uparrow	<code>\upuparrows</code>
\Downarrow	<code>\downdownarrows</code>	\Uparrow	<code>\upharpoonleft</code>
\Uparrow	<code>\upharpoonright</code>	\Downarrow	<code>\downharpoonright</code>
\rightsquigarrow	<code>\rightsquigarrow</code>	\leftrightsquigarrow	<code>\leftrightsquigarrow</code>

Tabela 3.15: AMS negirane binarne relacije in puščice.

\nless	\ngtr	\varsubsetneqq
\lneq	\gneq	\varsupsetneqq
\nleq	\ngeq	\nsubseteqeq
\nleqslant	\ngeqslant	\nsupseteqeq
\lneqq	\gneqq	\nmid
\lvertneqq	\gvertneqq	\nparallel
\nleqq	\ngeqq	\nshortmid
\lnsim	\gnsim	\nshortparallel
\lnapprox	\gnapprox	\nsim
\nprec	\nsucc	\ncong
\npreceq	\nsucceq	\nvdash
\nprecneqq	\nsuccneqq	\nvDash
\nprecnsim	\succnsim	\nVdash
\nprecnapprox	\succnapprox	\nVDash
\subsetneq	\supsetneq	\ntriangleleft
\varsubsetneqq	\varsupsetneqq	\ntriangleright
\nsubseteqeq	\nsupseteqeq	\ntrianglelefteq
\nsubseteqqq	\nsupseteqqq	\ntrianglerighteq
\nleftarrow	\rightarrow	\nleftrightarrow
\nLeftarrow	\Rightarrow	\nLeftrightarrow

Tabela 3.16: AMS binarne operacije.

$\dot{+}$	\cdot	\centerdot
\ltimes	\rtimes	\divideontimes
\doublecup	\doublecap	\smallsetminus
\veebar	$\bar{\wedge}$	\doublebarwedge
\boxplus	\boxminus	\circleddash
\boxtimes	\boxdot	\circledcirc
\intercal	\circledast	\rightthreetimes
\curlyvee	\curlywedge	\leftthreetimes

Tabela 3.17: AMS razni simboli.

\hbar	<code>\hbar</code>	\hbar	<code>\hslash</code>	\mathbb{k}	<code>\Bbbk</code>
\square	<code>\square</code>	\blacksquare	<code>\blacksquare</code>	\textcircled{S}	<code>\circledS</code>
\triangle	<code>\vartriangle</code>	\blacktriangle	<code>\blacktriangle</code>	\complement	<code>\complement</code>
∇	<code>\triangledown</code>	\blacktriangledown	<code>\blacktriangledown</code>	\Game	<code>\Game</code>
\diamond	<code>\lozenge</code>	\blacklozenge	<code>\blacklozenge</code>	\star	<code>\bigstar</code>
\sphericalangle	<code>\angle</code>	\sphericalangle	<code>\measuredangle</code>		
\diagup	<code>\diagup</code>	\diagdown	<code>\diagdown</code>	\backprime	<code>\backprime</code>
\nexists	<code>\nexists</code>	\Finv	<code>\Finv</code>	\varnothing	<code>\varnothing</code>
\eth	<code>\eth</code>	\sphericalangle	<code>\sphericalangle</code>	\mho	<code>\mho</code>

Tabela 3.18: Matematična abeceda.

Primer	Ukaz	Potreben paket
$ABCDEabcde1234$	<code>\mathrm{ABCDE abcde 1234}</code>	
$ABCDEabcde1234$	<code>\mathit{ABCDE abcde 1234}</code>	
$ABCDEabcde1234$	<code>\mathnormal{ABCDE abcde 1234}</code>	
$ABCDE$	<code>\mathcal{ABCDE abcde 1234}</code>	
$\mathcal{A}\mathcal{B}\mathcal{C}\mathcal{D}\mathcal{E}$	<code>\mathscr{ABCDE abcde 1234}</code>	<code>mathrsfs</code>
$\mathfrak{A}\mathfrak{B}\mathfrak{C}\mathfrak{D}\mathfrak{E}abcde1234$	<code>\mathfrak{ABCDE abcde 1234}</code>	<code>amsfonts</code> ali <code>amssymb</code>
$\mathbb{A}\mathbb{B}\mathbb{C}\mathbb{D}\mathbb{E}\mathbb{K}\mathbb{H}\mathbb{I}\mathbb{J}$	<code>\mathbb{ABCDE abcde 1234}</code>	<code>amsfonts</code> ali <code>amssymb</code>

Poglavje 4

Posebnosti

Pri sestavljanju obsežnega dokumenta nam \LaTeX pomaga z nekaterimi posebnostmi kot so npr. generiranje stvarnega kazala, urejanje seznama literature, in ostale stvari. Podrobnejši opis vsega, kar se da narediti z \LaTeX om lahko najdete v *LaTeX Manual* [1] in *The LaTeX Companion* [3].

4.1 Vključevanje EPS slik

Osnovna pripomočka v \LaTeX u za delo s plavajočimi objekti, kot so slike in tabele, sta okolji `figure` in `table`.

Obstajajo različne možnosti za risanje preprostih slik ali v osnovnem \LaTeX u ali pa s pomočjo kakšnega dodatnega paketa, nekaj od teh možnosti je predstavljenih v poglavju 5. Še več informacij lahko poiščete v *The LaTeX Companion* [3] in *LaTeX Manual* [1].

Preprostejši način za vključevanje slike v dokument je, da sliko narišemo s kakšnim izmed programov za risanje slik¹, potem pa dokončano sliko vključimo v dokument. Dodatni paketi za \LaTeX ponujajo različne možnosti za vključevanje slik, narejenih z drugimi programi. Mi bomo obravnavali le uporabo slik oblike Encapsulated PostScript (EPS), saj gre za dokaj preprost in zelo razširjen grafični format. Če želimo uporabljati slike v EPS obliki, potem potrebujemo PostScript tiskalnik².

Pripraven paket z dodatnimi ukazi za vključevanje slik je `graphicx`, katerega avtor je D. P. Carlisle. Spada v družino paketov imenovano »graphics«³.

Predpostavimo, da uporabljate sistem, kjer lahko izpisujete PostScript datoteke in da imate nameščen paket `graphicx`. Potem lahko sliko v dokument vključite z naslednjim postopkom:

-
1. npr. XFig, CorelDraw!, Freehand, Gnuplot, ...
 2. Druga možnost je, da uporabimo program GhostScript, ki je na voljo na `CTAN:/tex-archive/support/ghostscript`. V Windows in OS/2 okolju lahko uporabljamo tudi program GSview.
 3. `macros/latex/required/graphics`

1. Sliko shranite iz vašega programa za risanje v EPS obliki.⁴
2. V preambuli dokumenta vključite paket `graphicx` z ukazom

```
\usepackage[driver]{graphicx}
```

kjer je *driver* ime programa za konverzijo iz DVI v PostScript. Najpogostejši program za ta namen je `dvips`. Ime programa je obvezno potrebno podati, saj ne obstaja noben standard, po katerem se slike vključujejo v \TeX . Ker pozna ime programa *driver*, lahko paket `graphicx` izbere pravilno metodo za vnos podatkov o sliki v `.dvi` datoteko, tako da bo tiskalnik to razumel in pravilno vključil `.eps` datoteko.

3. Z ukazom

```
\includegraphics[opcija=vrednost, ...]{file}
```

vključimo datoteko *file* v dokument. V neobveznem parametru lahko podamo z vejicami ločen seznam opcij in ustreznih vrednosti. Z opcijami lahko spremenimo širino, višino in rotacijo slike, ki jo vključujemo. Najpomembnejše opcije so našteje v tabeli 4.1.

Tabela 4.1: Ključne opcije za paket `graphicx`.

<code>width</code>	sliko povečaj/zmanjšaj na predpisano širino
<code>height</code>	sliko povečaj/zmanjšaj na predpisano širino
<code>angle</code>	zavrti sliko za dani kot v pozitivni smeri
<code>scale</code>	za dani faktor povečaj/zmanjšaj sliko

Naslednji primer bo razjasnil zadeve:

```
\begin{figure}
\begin{center}
\includegraphics[angle=90, width=0.5\textwidth]{test}
\end{center}
\end{figure}
```

4. Če vaš program ne omogoča shranjevanja v EPS obliki, si lahko pomagate s tem, da namestite gonilnik za tiskalnik, ki tiska PostScript (npr. Apple LaserWriter), potem pa namesto tiskanja na ta tiskalnik izberete shranjevanje na datoteko. Z malce sreče bo ta datoteka imela EPS obliko. Pazite na to, da EPS, ki ga želite vključiti, nima več kot eno stran.

V tem primeru vključimo sliko shranjeno v datoteki `test.eps`. Sliko *najprej* zavrtimo za 90 stopinj, *nato* pa širino slike nastavimo na polovico širine standardnega odstavka. Razmerje med višino in širino ves čas ostaja nespremenjeno 1.0, saj višina ni predpisana. Širino in višino lahko določimo tudi v absolutnih dimenzijah. Za več podatkov pogledjte tabelo 6.5 na strani 109. Če želite izvedeti več kot v tem poglavju, potem - preberite [9] in [13].

4.2 Literatura

Seznam literature sestavljamo v okolju `thebibliography`. Vsak kos literature v seznamu se začne z

```
\bibitem{oznaka}
```

Parameter *oznaka* se uporablja za citiranje knjige ali članka znotraj dokumenta z ukazom

```
\cite{oznaka}
```

Če ne uporabite argumenta *oznaka*, je oštevilčenje del avtomatično. Dodatni parameter za ukazom `\begin{thebibliography}` je maksimalna širina oznake literature. V naslednjem primeru tako `{99}` za \LaTeX pomeni, da nobena oznaka literature ne bo širša od števila 99.

```
Partl~\cite{pa} has
proposed that \ldots
\begin{thebibliography}{99}
\bibitem{pa} H.~Partl:
\emph{German \TeX},
TUGboat Volume~9, Issue~1 (1988)
\end{thebibliography}
```

Partl [1] has proposed that ...

Literatura

[1] H. Partl: *German \TeX* , TUGboat
Volume 9, Issue 1 (1988)

Za večje projekte lahko uporabljate program Bib \TeX . Ta je vsebovan v večini distribucij \LaTeX a. Program omogoča vodenje baze celotnega seznama literature, iz katere lahko potegnemo reference, ki jih potrebujemo v članku. Bib \TeX generira sezname literature v skladu z različnimi predlogami, ki omogočajo izdelavo seznama literature na čisto poseben način.

Tabela 4.2: Zgledi vnosa gesla za stvarno kazalo.

Zgled	V kazalu	Komentar
<code>\index{hello}</code>	hello, 1	Navaden vnos
<code>\index{hello!Peter}</code>	Peter, 3	Podgeslo gesla ‘hello’
<code>\index{Sam@\textsl{Sam}}</code>	<i>Sam</i> , 2	Formatiran vnos
<code>\index{Lin@\textbf{Lin}}</code>	Lin , 7	Formatiran vnos
<code>\index{Jenny textbf}</code>	Jenny, 3	Formatirana oznaka strani
<code>\index{Joe textit}</code>	Joe, <i>5</i>	Formatirana oznaka strani

4.3 Stvarno kazalo

Zelo uporaben del številnih knjig je stvarno kazalo. V \LaTeX u lahko s pomožnim programom `makeindex`⁵, prav enostavno sestavimo stvarno kazalo. V teh navodilih bomo razložili osnovne ukaze za izdelavo stvarnega kazala, več podrobnosti pa lahko najdete v *The \LaTeX Companion* [3].

Da vključimo sestavljanje stvarnega kazala v \LaTeX u, moramo v preambuli dokumenta naložiti paket `makeidx` z ukazom:

```
\usepackage{makeidx}
```

in z ukazom

```
\makeindex
```

vključiti ukaze za sprotno beleženje podatkov za stvarno kazalo na posebno datoteko.

Stvarno kazalo sestavljamo z ukazom

```
\index{geslo}
```

kjer je *geslo* podatek za stvarno kazalo. Ta ukaz uporabimo na vseh mestih v dokumentu, na katere želimo, da kaže stvarno kazalo. Tabela 4.2 na različnih primerih pojasnjuje, kako vnašamo argument *geslo*.

Ko \LaTeX prevaja vhodno datoteko, ukaz `\index` ustrezno geslo skupaj s številko strani izpiše na posebno datoteko. Ta datoteko ima enako ime kot vhodna datoteka, a drugačno končnico (`.idx`). To `.idx` datoteko moramo

5. Na sistemih, ki ne podpirajo imen datotek, daljših od 8 črk, se program lahko imenuje `makeidx`.

potem obdelati s programom `makeindex`. Ustrezni ukaz v ukazni vrstici je

```
makeindex file
```

kjer je *file* ime vhodne datoteke za \LaTeX . Program `makeindex` združi enaka gesla za stvarno kazalo, gesla uredi po abecedi in zgradi datoteko z enakim imenom, a tokrat s končnico `.ind`, kjer so urejena gesla za stvarno kazalo. Sedaj se pri novem prevajanju vhodne datoteke to stvarno kazalo na `.ind` datoteki vključi na tistem mestu, kjer \LaTeX najde ukaz

```
\printindex
```

Datoteka `.ind` ima npr. obliko

```
\begin{theindex}
\item matrika 10,20
\subitem simetrična 15
\subsubitem pozitivno definitna 16
\subitem trikotna 17
\indexspace
\item vektor 7,17
\end{theindex}
```

Stvarno kazalo lahko s popraviljanjem `.ind` datoteke tudi ročno popravimo ali sestavimo. Na voljo imamo ukaze `\item`, `\subitem`, `\subsubitem` in `\indexspace`. To še posebno pride v poštev pri slovenščini, saj program `makeindex` šumnikov ne uredi pravilno. Pred zadnjim izpisom moramo zato vedno ročno pregledati datoteko s stvarnim kazalom, da bo pravilno urejena.

Paket `showidx`, ki je priložen \LaTeX 2 ϵ , izpiše vsa gesla za stvarno kazalo na mestih, kjer je ukaz `\index`, na levem robu teksta. To nam pri korekturah močno olajša preverjanje vsebine stvarnega kazala.

Bodite pozorni na to, da lahko ukaz `\index` vpliva na videz vašega dokumenta, če ga ne uporabljate pazljivo.

Moje geslo `\index{geslo}`. To ni isto kot `geslo\index{geslo}`. Poglejte položaje pik na koncu stavkov.

Moje geslo . To ni isto kot geslo. Poglejte položaje pik na koncu stavkov.

4.4 Prilagojene glave strani

Paket `fancyhdr`⁶, ki ga je napisal Piet van Oostrum, doda \LaTeX u nekaj preprostih ukazov, s katerimi lahko spremenimo glave in dna strani v svojih

6. Dosegljiv na CTAN:/tex-archive/macros/latex/contrib/supported/fancyhdr.

dokumentih. Primer možne uporabe tega paketa lahko vidite na vrhu te strani.

```

\documentclass{book}
\usepackage{fancyhdr}
\pagestyle{fancy}
% s tem poskrbimo, da so naslovi poglavij in razdelkov
% zapisani z malimi črkami.
\renewcommand{\chaptermark}[1]{\markboth{#1}{}}
\renewcommand{\sectionmark}[1]{\markright{\thesection\ #1}}
\fancyhf{} % zbrisemo trenutne nastavitve za header in footer
\fancyhead[LE,RO]{\bfseries\thepage}
\fancyhead[LO]{\bfseries\rightmark}
\fancyhead[RE]{\bfseries\leftmark}
\renewcommand{\headrulewidth}{0.5pt}
\renewcommand{\footrulewidth}{0pt}
\addtolength{\headheight}{0.5pt} % presledek za črto
\fancypagestyle{plain}{%
  \fancyhead{} % na straneh brez števil ni glave
  \renewcommand{\headrulewidth}{0pt} % in črte
}

```

Slika 4.1: Zgled uporabe paketa fancyhdr.

Pri prilagajanju glave in dna strani je ponavadi zapleteno vanje vključiti stvari, kot so npr. trenutni naslov poglavja in razdelka. V \LaTeX u to dosežemo z dvostopenjskim pristopom. V definiciji glave (header) in dna (footer) uporabimo ukaza `\rightmark` in `\leftmark`, ki predstavljata trenutni naslov razdelka oziroma poglavja. Vrednosti teh dveh ukazov se avtomatično spremenita vsakič ko \LaTeX naleti na ukaz za nov razdelek ali novo poglavje.

Da je stvar še bolj prilagodljiva, ukaz `\chapter` in sorodni ukazi ne definirajo znova samih ukazov `\rightmark` in `\leftmark`, temveč kličejo ukaze `\chaptermark`, `\sectionmark` ali `\subsectionmark`, ki so potem zadolženi za to, da ponovno definirajo ukaza `\rightmark` in `\leftmark`.

Torej, če želite spremeniti videz naslova poglavja v glavi, je najpreprosteje na novo definirati ukaz `\chaptermark`.

Slika 4.1 prikazuje možne nastavitve paketa `fancyhdr`, s katerimi glave in dna dobijo tak videz, kot ga imajo v tej knjigi. V vsakem primeru pa vam priporočam, da si pogledate še dokumentacijo za paket `fancyhdr`.

4.5 Paket verbatim

V tej knjigi smo že razložili, kako deluje *okolje verbatim*. V tem razdelku pa se bomo naučili uporabljati *paket verbatim*. Paket *verbatim* je v bistvu ponovna implementacija okolja *verbatim*, ki pa odpravi nekatere omejitve originalnega okolja *verbatim*. Samo po sebi to ni tako spektakularno, razlog, da to omenjamo, pa je nova uporabnost, ki jo imamo v paketu *verbatim* z ukazom

```
\verbatiminput{datoteka}
```

ki v dokument iz navedene datoteke vključi navaden ASCII tekst tako, kot če bi ga vnesli v okolju *verbatim*.

Ker je paket *verbatim* del svežnja ‘tools’, bi moral biti nameščen tudi na vašem sistemu. Več informacij o paketu lahko dobite v [10].

4.6 Prenos in namestitvev dodatnih L^AT_EX paketov

Večina namestitvev L^AT_EXa vključuje veliko dodatnih paketov, še veliko več pa jih je na voljo na internetu. Glavni naslov, kje lahko iščemo dodatne pakete je CTAN (<http://www.ctan.org/>).

Paketi, kot npr. *geometry*, *hyphenat* in ostali, so tipično sestavljeni iz dveh datotek: prva ima končnico *.ins*, druga pa končnico *.dtx*. Pogosto je zraven priložena še datoteka *readme.txt* s kratkim opisom paketa in navodili. Pred namestitvijo je potrebno to datoteko obvezno prebrati.

Kakorkoli že, ko enkrat na svoj računalnik prenesete datoteki *.dtx* in *.ins*, ju morate še vedno prevesti tako da bo (a) vaše T_EX okolje spoznalo nov paket in (b) da boste dobili dokumentacijo. Prvi del naredimo takole:

1. Poženemo L^AT_EX na datoteki *.ins*. Na ta način izluščimo datoteko *.sty*.
2. Datoteko *.sty* je potrebno prestaviti na mesto, kjer jo bo L^AT_EX našel. Ponavadi je to podimenik `... \localtexmf \tex \latex`.
3. Osvežimo L^AT_EXovo bazo podatkov o imenih datotek. To je odvisno od distribucije L^AT_EXa: teTeX, fpTeX – `texhash`; web2c – `maktexlsr`; MikTeX – `initexmf -update-fndb` ali pa uporabimo grafični vmesnik.⁷

Sedaj lahko iz datoteke *.dtx* izluščimo dokumentacijo:

⁷ Npr. če uporabljamo MikTeX v Windows okolju, potem poženemo program MikTeX Options in na prvi strani pritisnemo gumb Refresh now.

1. Poženemo \LaTeX na datoteki `.dtx`. To naredi `.dvi` datoteko. Da bodo vse reference pravilne, je potrebno \LaTeX pognati več kot enkrat.
2. Pogledamo, če je \LaTeX med ostalimi datotekami, ki so nastale, naredil tudi datoteko `.idx`. Če te datoteke ni, gremo lahko na korak 5.
3. Narediti moramo stvarno kazalo z ukazom:


```
makeindex -s gind.ist ime
```

 (kjer *ime* pomeni ime glavnega dokumenta brez končnice).
4. Še enkrat poženemo \LaTeX na `.dtx`.
5. Ne nazadnje, da bo branje navodil udobnejše, se spleča narediti `.ps` ali `.pdf` datoteko.

V nekaterih primerih \LaTeX naredi tudi datoteko `.glo` s slovarjem (glossary). V tem primeru med korakoma 4 in 5 v ukazni vrstici poženemo:

```
makeindex -s gglo.ist -o ime.gls ime.glo
```

Preden gremo na korak 5 moramo še zadnjič pognati \LaTeX na `.dtx` datoteki.

4.7 Uporaba orodja pdf \LaTeX

Avtor: Daniel Flipo <Daniel.Flipo@univ-lille1.fr>

PDF (Portable Document Format) je hipertekstovni format za dokumente. Podobno, kot na internetnih straneh, so lahko nekatere besede v dokumentu označene kot hiperpovezave, ki kažejo ali na druga mesta v dokumentu ali pa celo na druge dokumente. Če kliknemo na tako hiperpovezavo se znajdemo na drugi strani povezave. V okviru \LaTeX a vse pojavitve ukazov `\ref` in `\pageref` avtomatično postanejo hiperpovezave. Poleg tega se to avtomatično zgodi tudi s stvarnim kazalom, kazalom, in ostalimi podobnimi strukturami, ki se vse spremenijo v kolekcije hiperpovezav.

Večina strani na internetu je zapisanih v obliki HTML (*HyperText Markup Language*). Ta format ima dve pomembni slabosti za zapisovanje znanstvenih dokumentov:

1. Vključevanje matematičnih formul v HTML dokumente v splošnem ni podprto. Sicer res za to obstaja standard, a ga večina brskalnikov ne podpira oziroma nimajo vseh potrebnih pisav.
2. Izpisovanje HTML dokumentov je sicer možno, a se izpisi, dobljeni na različnih operacijskih sistemih oziroma različnih brskalnikih, lahko dokaj razlikujejo. Dobljeni rezultati so daleč od kvalitete, na katero smo se navadili pri uporabi \LaTeX a.

V preteklosti je bilo kar nekaj poskusov kreiranja prevajalnika iz \LaTeX a v HTML. Nekaj jih je bilo dokaj uspešnih v smislu da je z njimi možno

narediti veljavne internetne strani iz standardnih L^AT_EX vhodnih datotek. Toda vsi takšni prevajalniki ubirajo bližnjice, da dokončajo delo. Kakor hitro začnemo uporabljati zapletenejša orodja v L^AT_EXu ali dodatne pakete, imajo težave. Tako so se avtorji, ki so želeli tudi pri objavi na internetu zadržati kvaliteten videz svojih dokumentov, obrnili na PDF, ki ohranja videz dokumentov in dovoljuje navigacijo s hiperpovezavami. Večina sodobnih brskalnikov ima že vključeno podporo za direkten prikaz PDF dokumentov s pomočjo ustreznih dodatnih programov, kot je npr. Adobe Reader.

Čeprav skoraj za vsako računalniško okolje obstajajo pregledovalniki DVI in PS datotek, sta pregledovalnika PDF datotek Adobe Reader in Xpdf še bolj razširjena. Tako so dokumenti, ki so na voljo tudi v PDF obliki, na voljo večjemu številu potencialnih bralcev.

4.7.1 PDF dokumenti za svetovni splet

Kreiranje PDF datoteke iz L^AT_EX vhodnega dokumenta je zelo enostavno, saj imamo na voljo program pdfT_EX, ki ga je razvil Hàn Thê Thành. pdfT_EX naredi PDF datoteko podobno tako kot T_EX naredi DVI. Obstaja tudi pdfL^AT_EX, ki naredi PDF datoteko iz L^AT_EX vhodnega dokumenta.

Tako pdfT_EX kot pdfL^AT_EX se avtomatično namestita v večini sodobnih T_EX distribucij, kot so teT_EX, fpT_EX, MikT_EX, T_EXLive in CMacT_EX.

Če želimo namesto DVI narediti PDF dokument, potem je dovolj pri prevajanju zamenjati ukaz `latex file.tex` z ukazom `pdflatex file.tex`. Na sistemih, kjer se L^AT_EX ne poganja iz vhodne vrstice, je potrebno poiskati pravi gumb v T_EXControlCenter.

V L^AT_EXu lahko definiramo velikost strani z dodatnim argumentom pri klicu ukaza `\documentclass`, npr. z `a4paper` ali `letterpaper`. To deluje tudi v pdfL^AT_EXu, toda poleg tega mora pdfT_EX poznati tudi fizične mere papirja, da lahko določi fizične mere strani v PDF datoteki. Če uporabljate paket `hyperref` (poglejte stran 76), potem se velikost strani samodejno prilagodi. Sicer pa moramo to narediti sami, tako da v preambulo dokumenta vključimo naslednje vrstice:

```
\pdfpagewidth=\paperwidth
\pdfpageheight=\paperheight
```

V naslednjem razdelku bomo podrobneje razložili razliko med navadnim L^AT_EXom in pdfL^AT_EXom. Glavne razlike so na naslednjih področjih: uporabljene pisave, oblika slik, ki jih lahko vključimo in ročna konfiguracija povezav.

4.7.2 Pisave

pdfL^AT_EX zna delati z širokim naborom pisav (PK bitnimi pisavami, TrueType, POSTSCRIPT Type 1, ...) s tem da normalna L^AT_EXova pisava, bitna

PK pisava, zglada zelo grdo ko dokument odpremo s programom Adobe Reader. Če želimo narediti dokumente, ki bodo zgedali lepo, je najbolje uporabljati izključno pisavo POSTSCRIPT Type 1. *Sodobne namestitve TeXa imajo že nastavljene parametre tako, da se to zgodi samodejno. Najbolje je preveriti. Če deluje, potem lahko preskočite celoten razdelek.*

POSTSCRIPT Type 1 implementacijo pisav Computer Modern in AMS-Fonts so pripravili pri Blue Sky Research and Y&Y, Inc., ki so potem pravice prenesli na AMS (American Mathematical Society). Pisave so bile javno objavljene leta 1997 in so trenutno del vseh distribucij T_EXa.

Toda, če uporabljate L^AT_EX za pisanje dokumentov v jezikih, različnih od angleščine, potem najbrž želite uporabljati pisavo EC, LH, ali CB (preberi razpravo o pisavi OT1 na strani 27). Vladimir Volovich je pripravil paket cm-super pisav, ki pokriva celotne pisave EC/TC, EC Concrete, EC Bright in LH. Na voljo je na CTAN:/fonts/ps-type1/cm-super in je vključen v T_EXLive7 in MikT_EX. Podobno je type 1 CB grška pisava, ki jo je pripravil Apostolos Syropoulos, na voljo na CTAN:/tex-archive/fonts/greek/cb. Na žalost obe omenjeni seriji pisav nista tako kvalitetni kot pisave Type 1 CM, ki so jih pripravili pri Blue Sky/Y&Y, saj so pisave avtomatično rasterizirane, pri Type 1 CM pa je bilo to narejeno ročno, da je vsak znak optimalen. Čeprav dokumenti vseeno ne zgedajo tako lepo kot tisti, ki uporabljajo Type 1 CM pisave, dobimo na zaslonih z veliko ločljivostjo rezultate, ki so identični uporabi originalnih bitnih EC/LH/CB pisav.

Če pišemo dokumente v jeziku, ki uporablja latinico, imamo še nekaj dodatnih možnosti.

- Lahko uporabimo paket `aeguill`, kar naj bi pomenilo *skoraj evropski Computern Modern z dvojnimi srednjimi narekovaji* oz. *Almost European Computer Modern with Guillemets*. V preambulo dokumenta moramo vstaviti vrstico `\usepackage{aeguill}` in namesto EC pisav potem uporabljamo virtualne AE pisave.
- Alternativno lahko uporabljamo paket `mltex`, toda to deluje le v primeru, ko je bil pdfT_EX preveden z opcijo `mltex`.

AE virtualna pisava, prav tako pa tudi sistem M^IT_EX, prepriča T_EX, da ima na voljo pisavo s polnim naborom 256 znakov. To naredi tako, da manjkajoče znake kreira iz znakov CM pisave in jih preuredi v EC vrstnem redu. Tako lahko uporabljamo odlične Type 1 CM pisave, ki so na voljo na večini sistemov. Ker je pisava sedaj v kodiranju T1, bo deljenje delovalo pravilno za evropske jezike, ki uporabljajo latinico. Edina slabost tega pristopa je, da virtualni AE znaki ne delujejo s funkcijo `Find` v programu Adobe Reader, kar pomeni, da v končni PDF datoteki pri iskanju ne moremo uporabiti besed s šumniki.

Še ena rešitev je, da preklopimo na ostale POSTSCRIPT Type 1 pisave. Pravzaprav jih je nekaj celo vključenih v vsaki kopiji programa Adobe Re-

ader. Ker imajo te pisave drugačne velikosti znakov, se videz dokumenta spremeni. V splošnem te druge pisave uporabljajo večje presledke kot CM pisave, ki so glede presledkov zelo ekonomične pri porabi prostora. Spremeni se tudi vizualni videz dokumenta, saj pisave Times, Helvetica in Courier (glavni kandidati za uporabo pri zamenjavi) niso bile razvite tako, da bi lahko v sožitju nastopale v istem dokumentu.

V ta namen sta že pripravljeni dve množici pisav: `pxfonts`, ki temelji na pisavi *Palatino* in paket `txfonts`, ki temelji na pisavi *Times*. Za uporabo je dovolj v preambulo dokumenta vstaviti naslednje vrstice:

```
\usepackage[T1]{fontenc}
\usepackage{pxfonts}
```

Pozor: ko prevedete vašo vhodno datoteko, lahko v `.log` datoteki najdete vrstice v stilu

```
Warning: pdftex (file eurmo10): Font eur... not found
```

Te vrstice pomenijo, da nekatere pisave, ki smo jih uporabili v našem dokumentu, manjkajo. Te napake moramo odpraviti, sicer se lahko zgodi, da se v končnem PDF dokumentu *sploh ne prikažejo strani z manjkajočimi znaki*.

Kot vidite, se je mnogo ljudi obremenjevalo s pisavami, še posebno s tem, da ni nobene EC pisave, ki bi bila iste kvalitete kot CM Type 1 pisava. Pred kratkim se je pojavila nova množica visoko kvalitetnih pisav, ki se imenuje Latin Modern (LM). Ta je končala to nesrečno zgodbo. Če imate sodobno namestitev T_EXa, je velika verjetnost, da imate nameščene tudi LM pisave. V tem primeru moramo v preambulo dokumenta dodati

```
\usepackage{lmodern}
\usepackage[T1]{fontenc}
\usepackage{textcomp}
```

in dobili bomo odlične PDF dokumente, kjer so podprti vsi znaki iz latinice.

4.7.3 Uporaba zunanjih slik

Za vključevanje zunanjih slik v dokument je najbolje uporabiti paket `graphicx` (poglejte tudi stran 65). Če uporabimo posebno opcijo za gonilnik `pdftex`, bo paket deloval tudi s pdfL^AT_EXom:

```
\usepackage[pdftex]{color,graphicx}
```

V zgornjem primeru smo vključili tudi opcijo `color`, saj je povsem naravno, da v dokumentih, ki jih prikazujemo na spletu, uporabljamo tudi barve.

To je bila dobra novica. Slaba novica je, da slike v Encapsulated POSTSCRIPT formatu ne delujejo s PdfL^AT_EXom. Če ne definiramo končne datotek z ukazom `\includegraphics`, potem `graphicx` sam išče ustrezne

datoteke, pri čemer uporablja nastavitve izbranega gonilnika. V primeru `pdftex` so to formati `.png`, `.pdf`, `.jpg` in `.mps` (METAPOST)—*ne pa tudi .eps*.

Preprosta rešitev te težave je, da vse EPS datoteke pretvorimo v PDF obliko s pomočjo pomožnega programa `epstopdf`, ki se nahaja na številnih sistemih. Za vektorske slike je to dobra rešitev. Za bitne slike (fotografije, skenirani dokumenti) pa to ni idealno, saj PDF format podpira vključitev PNG in JPEG slik. PNG je dobra izbira za slike z zaslona in ostale slike z malo barvami, JPEG pa je dober za fotografije, saj ne porabi veliko prostora.

Včasih je tudi praktično, da določene geometrične oblike ne narišemo, temveč jo opišemo s posebnim ukaznim jezikom, kot je npr. METAPOST, ki je na voljo v večini distribucij T_EXa, priložen pa mu je zelo obsežen priročnik.

4.7.4 Hipertekstovne povezave

Paket `hyperref` poskrbi za to, da se vse interne reference znotraj dokumenta spremenijo v hiperpovezave. Da to deluje kot je treba, je potrebno malo čarovnije, zato moramo postaviti `\usepackage[pdftex]{hyperref}` kot *zadnji* ukaz v preambuli našega dokumenta.

Na voljo je mnogo opcij, s katerimi nastavimo obnašanje paketa `hyperref`: Podamo jih

- bodisi v obliki z vejicami ločenega seznama za opcijo `pdftex` v `\usepackage[pdftex]{hyperref}`
- ali v obliki posameznih vrstic z ukazom `\hypersetup{options}`.

Edina zares obvezna opcija je `pdftex`; ostale so neobvezne in z njimi lahko spreminjamo privzeto obnašanje paketa `hyperref`⁸ V naslednjem seznamu so privzete vrednosti zapisane s pokončno pisavo:

bookmarks (**=true,false**) Ali naj se med prikazom dokumenta levo prikaže seznam zaznamkov

unicode (**=false,true**) ali je v zaznamkih programa Adobe Reader možno uporabljati znake, ki ne spadajo v latinico

pdftoolbar (**=true,false**) ali naj se prikaže oz. skrije orodna vrstica v programu Adobe Reader

pdfmenubar (**=true,false**) ali naj se prikaže oz. skrije menu programa Adobe Reader

8. Pomembno je omeniti, da paket `hyperref` ni omejen le na pdfT_EX. Lahko ga uporabimo tudi za to, da v DVI izhodno datoteko, dobljeno z normalnim L^AT_EXom, vključimo še specifične PDF informacije. Te se potem zapišejo v PS datoteko ki jo kreiramo s programom `dvips` in jih na koncu uporabi program Adobe Distiller, če ga uporabimo, da nam PS datoteke spremeni v PDF.

- pdffitwindow** (**=true, false**) nastavi osnovno povečavo PDF, ko odpremo dokument
- pdftitle** (**=text**) nastavi naslov ki se prikaže v oknu `Document Info` v Adobe Readerju
- pdfauthor** (**=text**) ime avtorja dokumenta
- pdfnewwindow** (**=true, false**) ali naj se odpre novo okno kadar povezava kaže izven izbranega dokumenta
- colorlinks** (**=false, true**) ali naj bodo vse povezave uokvirjene z barvnimi okvirji (**false**) ali naj bo pobarvan tekst povezave (**true**). Barvo povezav lahko nastavimo z naslednjimi opcijami (prikazane so le privzete barve):
- linkcolor** (**=red**) barva notranje povezave (razdelek, stran, itd.),
 - citecolor** (**=green**) citat (povezava na seznam literature)
 - filecolor** (**=magenta**) barva povezave na datoteko
 - urlcolor** (**=cyan**) barva URL povezave (e-pošta, strani na splet)

Če smo zadovoljni s privzetimi vrednostmi, uporabimo le

```
\usepackage[pdftex]{hyperref}
```

Če želimo imeti odprt seznam zaznamkov in povezave v barvah (vrednosti **=true** ni obvezno pisati):

```
\usepackage[pdftex,bookmarks,colorlinks]{hyperref}
```

Kadar kreiramo PDF dokumente, ki jih bomo na koncu natisnili, pobarvane povezave niso najbolj primerne, saj bodo v končnem črno-belem izpisu sive in težko razvidne. Uporabimo lahko barvne okvirje, ki se ne natisnejo

```
\usepackage{hyperref}
\hypersetup{colorlinks=false}
```

ali pa so vse povezave črne:

```
\usepackage{hyperref}
\hypersetup{colorlinks,%
             citecolor=black,%
             filecolor=black,%
             linkcolor=black,%
             urlcolor=black,%
             pdftex}
```

Kadar želimo podati podatke za razdelek `Document Info` v PDF datoteki:

```
\usepackage[pdfauthor={Pierre Desproges},%
  pdftitle={Des femmes qui tombent},%
  pdftex]{hyperref}
```

Poleg avtomatičnih hiperpovezav med referencami, je možno vključiti tudi eksplisitne povezave z ukazom

```
\href{url}{text}
```

Naslednji stavek

```
The \href{http://www.ctan.org}{CTAN} website.
```

v izhodni datoteki prikaže »CTAN«; klik na besedo »CTAN« pa nas pripelje na spletno stran CTAN.

Kadar povezava ne kaže na spletno mesto temveč na lokalno datoteko, lahko uporabimo ukaz `\href` v obliki:

```
Celotna navodila so \href{manual.pdf}{tukaj}
```

ki prikaže tekst » Celotna navodila so [tukaj](#)«. Če kliknemo na besedo »[tukaj](#)« se odpre datoteka `manual.pdf`. (Ime datoteke je podano relativno glede na lokacijo trenutnega dokumenta).

Avtor članka bi mogoče rad, da bi mu bralci lahko preprosto poslali elektronsko pošto kar tako, da bi kliknili na pravo mesto v dokumentu. To lahko dosežemo z uporabo ukaza `\href` znotraj ukaza `\author` na naslovni strani dokumenta:

```
\author{Bor Plestenjak $<$
  \href{mailto:bor.plestenjak@fmf.uni-lj.si}%
  {bor.plestenjak@fmf.uni-lj.si}$>$}
```

Bodite pozorni na to, da sem povezavo vstavil tako, da se elektronski naslov ne pojavi samo v povezavi temveč tudi na sami strani. To sem naredil zato, ker bi povezava

```
\href{mailto:bor.plestenjak@fmf.uni-lj.si}{Bor Plestenjak}
```

delovala dobro znotraj Adobe Readerja, ko pa bi stran natisnili, povezava ne bi bila več vidna.

4.7.5 Težave s povezavami

Pri prevajanju lahko dobite sporočila podobna naslednjemu:

```
! pdfTeX warning (ext4): destination with the same
  identifier (name{page.1}) has been already used,
  duplicate ignored
```

To se zgodi, kadar se števec ponovno postavi na začetek, npr. ko uporabite ukaz `\mainmatter`, ki je na boljo v razredu `book`. Ta ukaz nastavi števec strani na 1 pred prvim poglavjem v knjigi. Toda, ker je v uvodu knjige tudi stran s številko 1, vse povezave na »stran 1« niso več enolične in zato se pojavi sporočilo »duplicate has been ignored.«

Protiukrep je, da med opcije paketa `hyperref` vstavimo `plainpages=false`. To na žalost pomaga le pri števcih strani. Še bolj radikalna rešitev je, da uporabimo opcijo `hypertexnames=false`, toda to bo povzročilo, da povezave na strani v stvarnem kazalu ne bodo več delovale.

4.7.6 Težave z zaznamki

Besedilo v zaznamkih ne zglada vedno tako, kot bi pričakovali. Ker so zaznamki le »navadno besedilo,« je za zaznamke na voljo manj črk in znakov kot za normalna besedila v L^AT_EXu. `Hyperref` bo normalno zaznal tovrstne težave in izpisal opozorilo:

```
Package hyperref Warning:
Token not allowed in a PDFDocEncoded string:
```

Ta problem lahko obidemo tako, da za zaznamek podamo navadno besedilo, ki nadomesti problematičen tekst:

```
\texorpdfstring{TEX tekst}{tekst za zaznamek}
```

Matematični izrazi so idealni kandidati za tovrstne težave:

```
\section{\texorpdfstring{ $E=mc^2$ }%
{E=mc^2}}
```

naredi iz `\section{ $E=mc^2$ }` zaznamek » $E=mc^2$ «.

Tudi sprememba barve pisave se ne razume najbolje z zaznamki:

```
\section{\textcolor{red}{Red !}}
```

vrne v zaznamku niz »redRed!«. Ukaz `\textcolor` se ignorira, izpiše pa se njegov argument `red`.

Boljše rezultate dobimo z ukazom

```
\section{\texorpdfstring{\textcolor{red}{Red !}}{Red\ !}}
```

Če pišemo dokument v kodiranju `unicode` in uporabimo v nastavitvah paketa `hyperref` opcijo `unicode`, potem lahko v zaznamkih uporabljamo vse znake, ki so na voljo v `unicode` kodni tabeli. Tako imamo na voljo veliko večji izbor znakov, ki jih lahko uporabimo v zaznamkih kot pri `\texorpdfstring`.

Združljivost L^AT_EX in pdfL^AT_EX dokumentov

Idealno bi se moral vsak dokument enako prevesti tako z L^AT_EX kot tudi s pdfL^AT_EX prevajalnikom. Glavna težava pri tem je vključevanje zunanjih slik. Preprosta rešitev je, da v *vseh* ukazih `\includegraphics` *sistematično opustimo* navajanje končnic datotek. Tako bo prevajalnik sam poiskal datoteko ustreznega formata na tekočem področju. V tem primeru moramo poskrbeti le za to, da imamo slike na voljo v ustreznih formatih za L^AT_EX in pdfL^AT_EX. Za L^AT_EX potrebujemo datoteke `.eps`, pdfL^AT_EX pa bo iskal datoteke s končnicami `.png`, `.pdf`, `.jpg` ali `.mps` (v navedenem vrstnem redu).

V primerih, ko bi radi uporabljali različno kodo za PDF verzijo dokumenta, lahko preprosto dodamo paket `ifpdf`⁹ v preambulo dokumenta. Velika verjetnost je, da imate ta paket nameščen, če pa ne in uporabljate MiK_TE_X, ga bo on namestil avtomatično prvič, ko ga boste potrebovali. V tem paketu je definiran poseben ukaz `\ifpdf` ki omogoča, da pišemo dokument s pogojno vsebino na preprost način. V naslednjem primeru hočemo, da bo PostScript verzija črno-bela da bo izpisovanje cenejše, PDF verzija, ki se bo uporabljala za prikaz na zaslonu pa naj bo barvna.

```

\RequirePackage{ifpdf} % ali poganjamo pdfTeX?
\ifpdf
  \documentclass[a4paper,12pt,pdftex]{book}
\else
  \documentclass[a4paper,12pt,dvips]{book}
\fi

\ifpdf
  \usepackage{lmodern}
\fi
\usepackage[bookmarks, % dodaj hiperpovezave
             colorlinks,
             plainpages=false]{hyperref}
\usepackage[T1]{fontenc}
\usepackage[latin1]{inputenc}
\usepackage[english]{babel}
\usepackage{graphicx}
...

```

V zgornjem primeru je vključen paket `hyperref` tudi v verzijo, ki ni PDF. Zaradi tega ukaz `\href` deluje v vseh primerih in ni potrebno pri vsaki

9. Če želite izvedeti celotno zgodbo o tem, zakaj uporabljati ta paket, si v seznamu pogostih vprašanj in odgovorov (FAQ) iz T_EXa oglejte <http://www.tex.ac.uk/cgi-bin/texfaq2html?label=ifpdf>.

pojavitvi ukaza uporabljati pogojni stavek.

V novejših distribucijah T_EXa (kot je npr. T_EXLive), se kot normalni T_EX program v resnici uporablja pdfT_EX, ki avtomatično predeluje dokumente v PDF in DVI obliko odvisno od nastavitvev v razredu dokumenta. Če pa uporabljamo zgornje stavke, potem jih s programom `pdflatex` prevedemo v PDF dokument, s programom `latex` pa v običajno DVI obliko.

4.8 Priprava elektronskih predstavitev

Avtor: Daniel Flipo <Daniel.Flipo@univ-lille1.fr>

Če želimo imeti predavanje, na katerem bi npr. predstavili svoje raziskovalne rezultate, lahko pišemo na tablo, uporabljamo prosojnice, ali pa uporabimo računalnik in ustrezní program za elektronske predstavitve.

pdfL^AT_EX v kombinaciji z razredom `beamer` omogoča pripravo predstavitev v obliki PDF, ki se obnašajo podobno če ne še boljše kot predstavitve narejene s programom PowerPoint, so pa veliko bolj prenosljive, saj za prikaz potrebujemo le Adobe Reader, ki je na voljo na bistveno več sistemih.

Razred `beamer` uporablja pakete `graphicx`, `color` in `hyperref` s parametri prilagojenimi predstavitvi na računalniškem zaslonu.

Ko prevedemo dokument, predstavljen na sliki 4.2 s `pdflatex`, dobimo PDF datoteko z naslovno stranjo in z drugo stranjo, ki prikazuje več reči, ki se odkrivajo po korakih ena za drugo, ko gremo skozi predstavitev.

Ena izmed prednosti razreda `beamer` je, da zna pri prevajanju direktno narediti PDF dokument z že kompletno pripravljeno predstavitvijo. Pri ostalih podobnih paketih pridemo do končne predstavitve v več fazah. Tako moramo pri uporabi paketa `prospcr` najprej pripraviti PostScript različico, pri paketu `ppower4` pa moramo dobljeno PDF datoteko še naknadno obdelati z dodatnim programom.

Z razredom `beamer` lahko na različne načine pripravimo dokument iz iste vhodne datoteke. Vhodna datoteka za te različne načine vsebuje posebna navodila v poševnih oklepajih. Na voljo so naslednji načini.

beamer za predstavitve v PDF obliki, kot smo jih predstavili pred tem,

trans za prosojnice,

handout za izpiske.

Privzeti način je `beamer`, Spremenimo ga lahko tako, da navedemo drugi način kot globalni parameter razreda `beamer`, kot npr.

```
\documentclass[10pt,handout]{beamer}
```

če želimo izpisati predavanje na papir.

Videz predstavitve na zaslonu je odvisen od izbrane teme. Izberemo lahko eno izmed številnih tem, ki so že vključene v razred `beamer` ali pa

```

\documentclass[10pt]{beamer}
\mode<beamer>{%
  \usetheme[hideothersubsections,
            right,width=22mm]{Goettingen}
}

\title{Preprosta predstavitev}
\author[D. Flipo]{Daniel Flipo}
\institute{U.S.T.L. \& GUTenberg}
\titlegraphic{\includegraphics[width=20mm]{USTL}}
\date{2005}

\begin{document}

\begin{frame}<handout:0>
  \titlepage
\end{frame}

\section{Zgled}

\begin{frame}
  \frametitle{Kaj lahko počnemo v primeru lepega vremena}
  \begin{block}{Lahko\ldots}
    \begin{itemize}
      \item peljemo psa na sprehod\ldots \pause
      \item beremo knjigo\pause
      \item se igramo z mačko\pause
    \end{itemize}
  \end{block}
  in še mnogo drugih stvari
\end{frame}
\end{document}

```

Slika 4.2: Preprost zgled za razred beamer

sami pripravimo novo temo. Za več podatkov o tem pogledjte obsežno dokumentacijo `beameruserguide.pdf`, ki je priložena razredu.

Podrobneje si pogledjmo ukaze uporabljene v vhodni datoteki na sliki 4.2.

Za način `\mode<beamer>` za predstavitev na zaslonu smo izbrali temo *Goettingen*, ki prikazuje tudi navigacijsko ploščo integrirano v seznam vsebine. S parametri lahko nastavimo širino plošče (v našem primeru 22 mm) in njen položaj (na desni strani glavnega besedila). Opcija `hideothersubsections` prikaže naslove poglavij in razdelke trenutnega poglavja. V primeru načinov `\mode<trans>` in `\mode<handout>` ni nobenih posebnih nastavitvev, zato se pojavita v standardni privzeti obliki.

Ukazi `\title{}`, `\author{}`, `\institute{}`, in `\titlegraphic{}` nastavijo vsebino naslovne strani. Z dodatnimi argumenti lahko pri ukazih `\title[]{} in \author[]{} podamo še verzijo naslova in imena avtorja, ki se prikazujeta na plošči v temi Goettingen.`

Naslovi in podnaslovi v plošči so narejeni z običajnimi ukazi `\section{}` in `\subsection{}`, ki jih postavimo *izven* okolja `frame`.

Z majhnimi ikonami za navigacijo na dnu zaslona se lahko premikamo skozi dokument. Pojav ikon je neodvisen od izbrane teme.

Vsebino vsakega zaslona (ene prosojnice) moramo postaviti v okolje `frame`. Z dodatnim argumentom v poševnih oklepajih (`< in >`) lahko zaslon izločimo iz enega izmed načinov predstavitve. V zgledu se tako zaradi uporabljenega argumenta `<handout:0>` prva stran ne pojavi na izpiskih.

Priporočljivo je, da naslov za vsako prosojnico nastavimo ločeno od naslovne prosojnice. To naredimo z ukazom `\frametitle{}`. Če potrebujemo še podnaslov, lahko uporabimo okolje `block`, kot je prikazano v zgledu. Bodite pozorni na to, da ukazi za razporejanje teksta v enote, kot sta npr. `\section{}` in `\subsection{}`, ne izpišejo pravilnega izpisa na prosojnico.

Z ukazom `\pause`, ki je v zgledu uporabljen v okolju `itemize`, lahko prikažemo vrstice eno po eno. Za ostale uporabne učinke, ki jih uporabljamo pri predstavitev, si pogledjte še delovanje ukazov `\only`, `\uncover`, `\alt` in `\temporal`. Na številnih mestih lahko z dodatnimi argumenti med poševnimi oklepaji še natančneje prilagodimo predstavitev našim željam.

Če želite dobiti dober pregled vseh možnosti, ki so na voljo v razredu `beamer`, preberite dokumentacijo `beameruserguide.pdf`. Paket se aktivno razvija, zato za zadnje informacije preverite tudi njegovo domačo stran <http://latex-beamer.sourceforge.net/>.

Poglavje 5

Priprava matematičnih slik

Večina uporabnikov z \LaTeX om ureja besedila. Čeprav \LaTeX ov pristop k urejanju besedil poudarja strukturo pred vsebino, \LaTeX vseeno ponuja nekaj zelo omejenih možnosti za prikaz grafičnih vsebin, ki jih opišemo z ukazi v vhodni datoteki. Obstaja veliko razširitev \LaTeX a, ki premostijo te omejitve in omogočajo še lažjo pripravo slik. Nekaj jih bomo predstavili v tem poglavju.

5.1 Kratki pregled

V okolju `picture` lahko slike programiramo direktno v \LaTeX u. Podroben opis lahko poiščete v *\LaTeX Manual* [1]. Na eni strani smo v tem okolju zelo omejeni z nekaterimi pogoji, tako so npr. nakloni daljic in polmeri krogov omejeni na majhen izbor možnih vrednosti. Na drugi strani pa imamo v okolju `picture` v $\text{\LaTeX} 2_{\epsilon}$ na voljo ukaz `\qbezier` za Bézierove krivulje, kjer »q« pomeni «kvadratna» (quadratic). Številne pogosto uporabljene krivulje kot so krogi, elipse, verižnice in ostale, se da dovolj dobro aproksimirati s kvadratnimi Bézierovimi krivuljami, res pa je, da za to včasih potrebujemo malo računanja. Za pripravo slik iz ukazov `\qbezier` lahko uporabimo tudi programski jezik kot je npr. Java in v tem primeu postane okolje `picture` zelo zmogljivo.

Čeprav je zapisovanje slike direktno v \LaTeX dokaj omejeno in dostikrat tudi zelo utrudljivo, še vedno obstajajo razlogi, da se slik lotimo na ta način. Tako dobljeni dokumenti zavzamejo manj spomina kot če vključujemo zunanje slike, pa še nobenih dodatnih datotek s slikami ne potrebujemo.

Paketi kot so `epic`, `eepic` (ta dva sta med drugim opisana v *The \LaTeX Companion* [3]) in `pstricks` odpravljajo omejitve originalnega okolja `picture` in močno povečajo grafične sposobnosti \LaTeX a.

Med tem ko prva dva prej omenjena paketa le razširita okolje `picture`, ima paket `pstricks` svoje okolje `pspicture`. Moč paketa `pstricks` izvira iz dejstva, da se v paketu v veliki meri uporabljajo možnosti, ki jih ponuja POSTSCRIPT. Poleg tega obstajajo še številni paketi za posebne potrebe.

En izmed teh je `Xy-pic`, ki ga bomo predstavili na koncu tega poglavja. Širok izbor teh paketov je podrobno predstavljen v *The L^AT_EX Graphics Companion* [4] (ne zamenjajte z *The L^AT_EX Companion* [3]).

Najmočnejše grafično orodje, povezano z L^AT_EXom, je najbrž `MetaPost`, dvojček Donald E. Knuthovega orodja `METAFONT`. `MetaPost` vsebuje zelo zmogljiv in matematično prefinjen programski jezik tako kot `METAFONT`. Razlika je, da `METAFONT` generira bitne slike, `MetaPost` pa generira slike v formatu Encapsulated POSTSCRIPT, ki jih lahko vključimo v L^AT_EX. Za uvod v `MetaPost` pogledjte *A User's Manual for MetaPost* [15] ali navodila na [17].

Zelo temeljito razpravo o strategijah za vključevanje slik (in pisav) v L^AT_EX in T_EX lahko najdete v *T_EX Unbound* [16].

5.2 Okolje `picture`

Avtor: Urs Oswald <osurs@bluewin.ch>

5.2.1 Osnovni ukazi

Okolje `picture`¹ vključimo z enim izmed naslednjih dveh ukazom

```
\begin{picture}(x,y)...\end{picture}
```

ali

```
\begin{picture}(x,y)(x_0,y_0)...\end{picture}
```

Števila x , y , x_0 , y_0 se nanašajo na količino `\unitlength`, ki jo lahko spremenimo kadarkoli (a ne znotraj okolja `picture`) z ukazom v stilu

```
\setlength{\unitlength}{1.2cm}
```

Privzeta vrednost za `\unitlength` je `1pt`. Prvi par, (x, y) , pove, koliko pravokotnega prostora znotraj dokumenta je potrebno rezervirati za sliko, neobvezni drugi par, (x_0, y_0) , pa nastavi te koordinate v spodnji levi kot rezerviranega pravokotnika.

1. Če verjamete ali ne, okolje `picture` deluje v povsem navadnem L^AT_EX 2_ε in zanj ni potrebno naložiti nobenega paketa.

Večina ukazov za risanje ima eno izmed naslednjih dve oblik

```
\put (x, y) {object}
```

ali

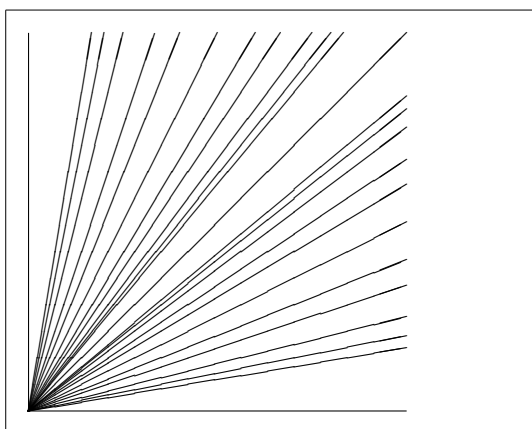
```
\multiput (x, y) (\Delta x, \Delta y) {n} {object}
```

Izjema so Bézierove krivulje, ki jih narišemo z ukazom

```
\qBezier (x1, y1) (x2, y2) (x3, y3)
```

5.2.2 Daljice

```
\setlength{\unitlength}{5cm}
\begin{picture}(1,1)
  \put(0,0){\line(0,1){1}}
  \put(0,0){\line(1,0){1}}
  \put(0,0){\line(1,1){1}}
  \put(0,0){\line(1,2){.5}}
  \put(0,0){\line(1,3){.3333}}
  \put(0,0){\line(1,4){.25}}
  \put(0,0){\line(1,5){.2}}
  \put(0,0){\line(1,6){.1667}}
  \put(0,0){\line(2,1){1}}
  \put(0,0){\line(2,3){.6667}}
  \put(0,0){\line(2,5){.4}}
  \put(0,0){\line(3,1){1}}
  \put(0,0){\line(3,2){1}}
  \put(0,0){\line(3,4){.75}}
  \put(0,0){\line(3,5){.6}}
  \put(0,0){\line(4,1){1}}
  \put(0,0){\line(4,3){1}}
  \put(0,0){\line(4,5){.8}}
  \put(0,0){\line(5,1){1}}
  \put(0,0){\line(5,2){1}}
  \put(0,0){\line(5,3){1}}
  \put(0,0){\line(5,4){1}}
  \put(0,0){\line(5,6){.8333}}
  \put(0,0){\line(6,1){1}}
  \put(0,0){\line(6,5){1}}
\end{picture}
```



Daljico narišemo z ukazom

```
\put(x,y){\line(x1,y1){length}}
```

Ukaz `\line` ima dva argumenta:

1. smerni vektor,
2. dolžino.

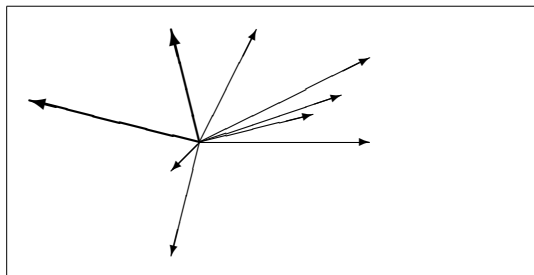
Vrednosti komponent v smernem vektorju so omejene na cela števila

$$-6, -5, \dots, 5, 6,$$

komponenti pa morata biti tuji števili (razen 1 nimata nobenega skupnega delitelja). Na sliki je predstavljenih vseh 25 možnih naklonov v prvem kvadrantu. Dolžina je podana relativno glede na `\unitlength`. V primeru navpične daljice kot dolžino podamo navpično koordinato, v vseh drugih pa vodoravno koordinato konca daljice.

5.2.3 Puščice

```
\setlength{\unitlength}{0.75mm}
\begin{picture}(60,40)
  \put(30,20){\vector(1,0){30}}
  \put(30,20){\vector(4,1){20}}
  \put(30,20){\vector(3,1){25}}
  \put(30,20){\vector(2,1){30}}
  \put(30,20){\vector(1,2){10}}
  \thicklines
  \put(30,20){\vector(-4,1){30}}
  \put(30,20){\vector(-1,4){5}}
  \thinlines
  \put(30,20){\vector(-1,-1){5}}
  \put(30,20){\vector(-1,-4){5}}
\end{picture}
```



Puščico narišemo z ukazom

```
\put(x,y){\vector(x1,y1){length}}
```

Pri puščicah so komponente smernega vektorja še bolj omejene kot pri daljicah, saj jih lahko izbiramo le med celimi števili

$$-4, -3, \dots, 3, 4.$$

Tako kot pri daljicah morata komponenti biti tuji števili. Bodite pozorni na vpliv ukaza `\thicklines` na dve puščici, ki kažeta v zgornji levi kot.

5.2.4 Krožnice

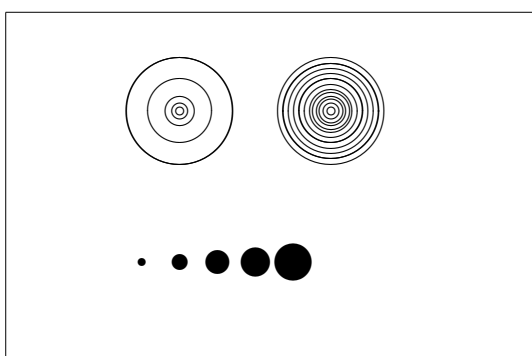
```

\setlength{\unitlength}{1mm}
\begin{picture}(60, 40)
  \put(20,30){\circle{1}}
  \put(20,30){\circle{2}}
  \put(20,30){\circle{4}}
  \put(20,30){\circle{8}}
  \put(20,30){\circle{16}}
  \put(20,30){\circle{32}}

  \put(40,30){\circle{1}}
  \put(40,30){\circle{2}}
  \put(40,30){\circle{3}}
  \put(40,30){\circle{4}}
  \put(40,30){\circle{5}}
  \put(40,30){\circle{6}}
  \put(40,30){\circle{7}}
  \put(40,30){\circle{8}}
  \put(40,30){\circle{9}}
  \put(40,30){\circle{10}}
  \put(40,30){\circle{11}}
  \put(40,30){\circle{12}}
  \put(40,30){\circle{13}}
  \put(40,30){\circle{14}}

  \put(15,10){\circle*{1}}
  \put(20,10){\circle*{2}}
  \put(25,10){\circle*{3}}
  \put(30,10){\circle*{4}}
  \put(35,10){\circle*{5}}
\end{picture}

```



Ukaz

```
\put( $x$ ,  $y$ ){\circle{ $diameter$ }}
```

nariše krožnico s središčem v točki (x, y) in premerom $diameter$. V okolju `picture` lahko uporabljamo le premere do približno 14 mm, s tem da pod to vrednostjo niso dopustni vse premeri. Ukaz `\circle*` nariše zapolnjeni krog.

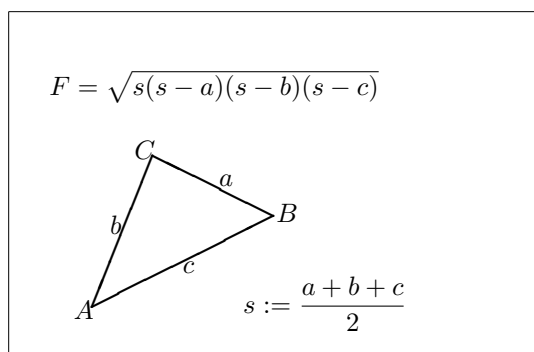
Tako kot pri daljicah se moramo, če želimo premer, ki ni podprt, obrniti na dodatne pakete, kot sta `eepic` in `pstricks`. Temeljite opis teh dveh paketov lahko najdete v *The L^AT_EX Graphics Companion* [4].

Obstaja pa tudi možnost znotraj okolja `picture`. Če lahko opravite potrebne izračune (oziroma jih prepustite ustreznemu programu), potem lahko poljubne daljice in krožnice sestavite iz kvadratnih Bézierovih krivulj.

Poglejte *Graphics in L^AT_EX 2_ε* [17] za primere in kode pomožnih programov v Javi.

5.2.5 Besedila in formule

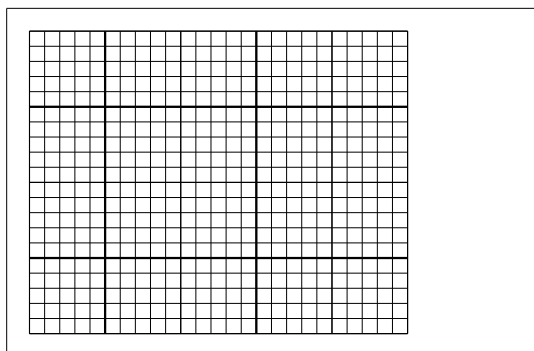
```
\setlength{\unitlength}{0.8cm}
\begin{picture}(6,5)
  \thicklines
  \put(1,0.5){\line(2,1){3}}
  \put(4,2){\line(-2,1){2}}
  \put(2,3){\line(-2,-5){1}}
  \put(0.7,0.3){$A$}
  \put(4.05,1.9){$B$}
  \put(1.7,2.95){$C$}
  \put(3.1,2.5){$a$}
  \put(1.3,1.7){$b$}
  \put(2.5,1.05){$c$}
  \put(0.3,4){$F=$
  \sqrt{s(s-a)(s-b)(s-c)}$}
  \put(3.5,0.4){$\displaystyle
  s:=\frac{a+b+c}{2}$}
\end{picture}
```



Kot prikazuje ta zgled, lahko besedilo in formule vstavimo v okolje picture s ukazom \put na povsem običajni način.

5.2.6 \multiput in \linethickness

```
\setlength{\unitlength}{2mm}
\begin{picture}(30,20)
  \linethickness{0.075mm}
  \multiput(0,0)(1,0){26}%
  {\line(0,1){20}}
  \multiput(0,0)(0,1){21}%
  {\line(1,0){25}}
  \linethickness{0.15mm}
  \multiput(0,0)(5,0){6}%
  {\line(0,1){20}}
  \multiput(0,0)(0,5){5}%
  {\line(1,0){25}}
  \linethickness{0.3mm}
  \multiput(5,0)(10,0){2}%
  {\line(0,1){20}}
  \multiput(0,5)(0,10){2}%
  {\line(1,0){25}}
\end{picture}
```



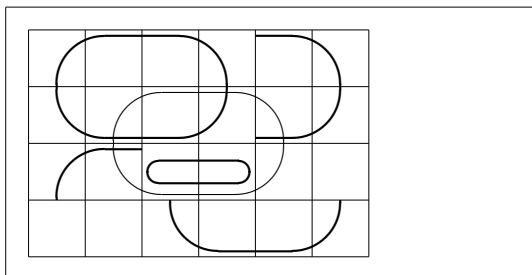
Ukaz

```
\multiput(x,y)(\Delta x,\Delta y){n}{object}
```

ima 4 argumente: začetno točko, vektor premika od enega objekta do naslednjega, število objektov in objekt, ki naj se nariše. Ukaz `\linethickness` vpliva na vodoravne in navpične daljice, ne pa na poševne daljice in krožnice. Vpliva pa, presenetljivo, na kvadratne Bézierove krivulje!

5.2.7 Ovali

```
\setlength{\unitlength}{0.75cm}
\begin{picture}(6,4)
  \linethickness{0.075mm}
  \multiput(0,0)(1,0){7}%
    {\line(0,1){4}}
  \multiput(0,0)(0,1){5}%
    {\line(1,0){6}}
  \thicklines
  \put(2,3){\oval(3,1.8)}
  \thinlines
  \put(3,2){\oval(3,1.8)}
  \thicklines
  \put(2,1){\oval(3,1.8)[t1]}
  \put(4,1){\oval(3,1.8)[b]}
  \put(4,3){\oval(3,1.8)[r]}
  \put(3,1.5){\oval(1.8,0.4)}
\end{picture}
```



Ukaz

```
\put(x,y){\oval(w,h)}
```

ali v obliki

```
\put(x,y){\oval(w,h)[polozaj]}
```

vrne oval s središčem v (x, y) , s širino w in z višino h . Neobvezni argument *polozaj* ima lahko vrednosti **b**, **t**, **l**, **r** ki po vrsti pomenijo zgoraj, spodaj, levo, desno, lahko pa jih tudi kombiniramo, kot je prikazano v zgledu.

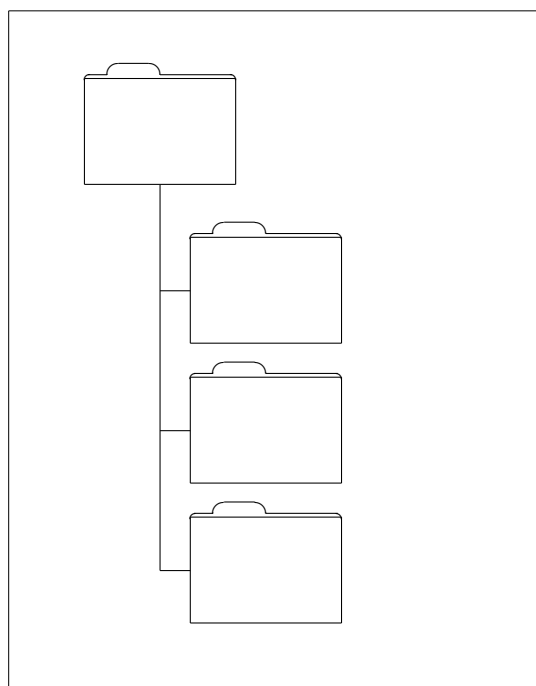
5.2.8 Debelina črt

Na debelino črt lahko vplivamo z dvema razredoma ukazov. Na eni strani je `\linethickness{length}` na drugi strani pa sta ukaza `\thinlines` in `\thicklines`. Medtem kot `\linethickness{length}` vpliva le na vodoravne

in navpične črte (in na kvadratne Bézierove krivulje), ukaza `\thinlines` in `\thicklines` vplivata na poševne črte, krožnice in ovale.

5.2.9 Večkratna uporaba vnaprej pripravljenih škatel s slikami

```
\setlength{\unitlength}{0.5mm}
\begin{picture}(120,168)
\newsavebox{\foldera}
\savebox{\foldera}
  (40,32)[bl]{% definicija
  \multiput(0,0)(0,28){2}
    {\line(1,0){40}}
  \multiput(0,0)(40,0){2}
    {\line(0,1){28}}
  \put(1,28){\oval(2,2)[t1]}
  \put(1,29){\line(1,0){5}}
  \put(9,29){\oval(6,6)[t1]}
  \put(9,32){\line(1,0){8}}
  \put(17,29){\oval(6,6)[tr]}
  \put(20,29){\line(1,0){19}}
  \put(39,28){\oval(2,2)[tr]}
  }
\newsavebox{\folderb}
\savebox{\folderb}
  (40,32)[l]{% definicija
  \put(0,14){\line(1,0){8}}
  \put(8,0){\usebox{\foldera}}
  }
\put(34,26){\line(0,1){102}}
\put(14,128){\usebox{\foldera}}
\multiput(34,86)(0,-37){3}
  {\usebox{\folderb}}
\end{picture}
```



Škatlo s sliko lahko *napovemo* z ukazom

```
\newsavebox{ime}
```

potem jo *definiramo* z

```
\savebox{ime}(width,height)[position]{content}
```

na koncu pa jo lahko *narišemo* kolikor krat želimo z

```
\put(x,y)\usebox{ime}
```

Dodatno neobvezni parameter *pozicija* definira mesto, na katerem je re-

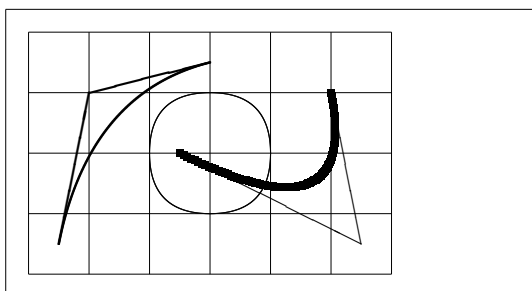
ferenčna točka shranjene škatle s sliko, oziroma mesto, kjer je škatla »zasidrana«. V zgornjem primeru je to mesto nastavljeno na `bl` kar postavlja mesto priveza v gornji levi kot škatle. Ostale možno vrednosti za ta parameter so še `t` (zgoraj) in `r` (desno).

Argument *ime* se nanaša na L^AT_EXov pomnilnik in ima naravo ukaza (zato v zgornjem primeru imena vsebujejo tudi poševnice). Škatle s slikami lahko vlagamo v druge škatle, v zgornjem primeru je tako npr. `\foldera` uporabljen v definiciji `\folderb`.

V definiciji smo morali uporabiti ukaz `\oval`, saj ukaz `\line` ne podpira daljic, kjer je dolžina segmenta manjša od 3 mm.

5.2.10 Kvadratne Bézierove krivulje

```
\setlength{\unitlength}{0.8cm}
\begin{picture}(6,4)
  \linethickness{0.075mm}
  \multiput(0,0)(1,0){7}
    {\line(0,1){4}}
  \multiput(0,0)(0,1){5}
    {\line(1,0){6}}
  \thicklines
  \put(0.5,0.5){\line(1,5){0.5}}
  \put(1,3){\line(4,1){2}}
  \qbezier(0.5,0.5)(1,3)(3,3.5)
  \thinlines
  \put(2.5,2){\line(2,-1){3}}
  \put(5.5,0.5){\line(-1,5){0.5}}
  \linethickness{1mm}
  \qbezier(2.5,2)(5.5,0.5)(5,3)
  \thinlines
  \qbezier(4,2)(4,3)(3,3)
  \qbezier(3,3)(2,3)(2,2)
  \qbezier(2,2)(2,1)(3,1)
  \qbezier(3,1)(4,1)(4,2)
\end{picture}
```



Kot prikazuje ta primer, ni dovolj dobro, če krožnico razdelimo le na 4 kvadratne Bézierove krivulje. Potrebujemo jih najmanj 8. Na sliki vidimo tudi, kakšne je vpliv ukazov `\linethickness` na vodoravne in navpične preme in kako `\thinlines` in `\thicklines` vplivata na poševne daljice. Vidimo lahko tudi, da oba razreda ukazov delujeta na kvadratne Bézierove krivulje, kjer vsak nov ukaz povozi prejšnjega.

Naj $P_1 = (x_1, y_1)$, $P_2 = (x_2, y_2)$ označujeta končni točki, in naj bosta m_1 in m_2 naklona kvadratne Bézierove krivulje v teh točkah. Vmesna kontrolna

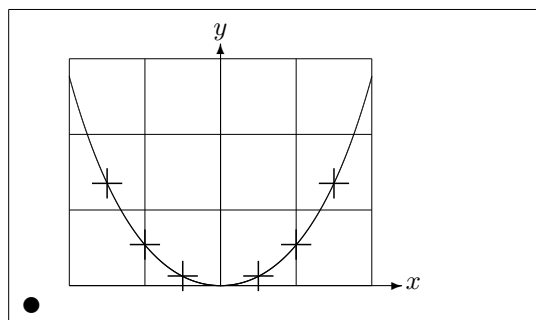
točka $S = (x, y)$ je potem podana z enačbami

$$\begin{cases} x = \frac{m_2 x_2 - m_1 x_1 - (y_2 - y_1)}{m_2 - m_1}, \\ y = y_i + m_i(x - x_i) \quad (i = 1, 2). \end{cases} \quad (5.1)$$

Poglejte *Graphics in L^AT_EX 2_ε* [17] za program v Javi, ki generira vrstice s potrebnimi ukazi `\qbezier`.

5.2.11 Verižnica

```
\setlength{\unitlength}{1cm}
\begin{picture}(4.3,3.6)(-2.5,-0.25)
\put(-2,0){\vector(1,0){4.4}}
\put(2.45,-.05){\text{\$x\$}}
\put(0,0){\vector(0,1){3.2}}
\put(0,3.35){\makebox(0,0){\text{\$y\$}}}
\qbezier(0.0,0.0)(1.2384,0.0)
(2.0,2.7622)
\qbezier(0.0,0.0)(-1.2384,0.0)
(-2.0,2.7622)
\linethickness{.075mm}
\multiput(-2,0)(1,0){5}
{\line(0,1){3}}
\multiput(-2,0)(0,1){4}
{\line(1,0){4}}
\linethickness{.2mm}
\put(.3,.12763){\line(1,0){.4}}
\put(.5,-.07237){\line(0,1){.4}}
\put(-.7,.12763){\line(1,0){.4}}
\put(-.5,-.07237){\line(0,1){.4}}
\put(.8,.54308){\line(1,0){.4}}
\put(1,.34308){\line(0,1){.4}}
\put(-1.2,.54308){\line(1,0){.4}}
\put(-1,.34308){\line(0,1){.4}}
\put(1.3,1.35241){\line(1,0){.4}}
\put(1.5,1.15241){\line(0,1){.4}}
\put(-1.7,1.35241){\line(1,0){.4}}
\put(-1.5,1.15241){\line(0,1){.4}}
\put(-2.5,-0.25){\circle*{0.2}}
\end{picture}
```



Na tej sliki je vsaka simetrična polovica grafa verižnice $y = \cosh x - 1$ aproksimirana s kvadratno Bézierovo krivuljo. Desna polovica krivulje se konča v točki $(2, 2.7622)$, kjer ima odvod vrednost $m = 3.6269$. S pomočjo enačb (5.1) lahko izračunamo vmesne kontrolne točke. Za njiju se izkaže, da sta $(1.2384, 0)$ in $(-1.2384, 0)$. Križi označujejo točke *prave* verižnice. Vidimo, da razlika skoraj ni opazna in je povsod pod enim procentom.

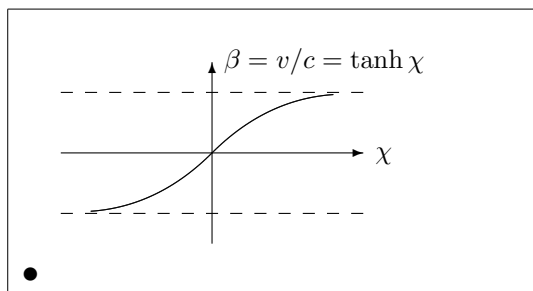
Ta primer kaže, kako lahko uporabljamo neobvezni argument v ukazu `\begin{picture}`. Slika je definirana v smiselni matematični koordinati, medtem ko ukaz

```
\begin{picture}(4.3,3.6)(-2.5,-0.25)
```

za njen spodnji levi kot (označen z črnim pobarvanim diskom) privzame koordinate $(-2.5, -0.25)$.

5.2.12 Hitrost v posebni teoriji relativnosti

```
\setlength{\unitlength}{0.8cm}
\begin{picture}(6,4)(-3,-2)
  \put(-2.5,0){\vector(1,0){5}}
  \put(2.7,-0.1){\chi}
  \put(0,-1.5){\vector(0,1){3}}
  \multiput(-2.5,1)(0.4,0){13}
    {\line(1,0){0.2}}
  \multiput(-2.5,-1)(0.4,0){13}
    {\line(1,0){0.2}}
  \put(0.2,1.4)
    {\beta=v/c=\tanh\chi}
  \qBezier(0,0)(0.8853,0.8853)
    (2,0.9640)
  \qBezier(0,0)(-0.8853,-0.8853)
    (-2,-0.9640)
  \put(-3,-2){\circle*{0.2}}
\end{picture}
```



Kontrolne točke dveh Bézierovih krivulj so bile izračunane s pomočjo formul (5.1). Pozitivna veja je določena s $P_1 = (0, 0)$, $m_1 = 1$ in $P_2 = (2, \tanh 2)$, $m_2 = 1/\cosh^2 2$. Slika je definirana z matematično najustrežnejšimi koordinatami, potem pa levemu spodnjemu kotu določimo koordinate $(-3, -2)$ (črn krog).

5.3 Xy-pic

Avtor: Alberto Manuel Brandão Simões <albie@alfarrabio.di.uminho.pt>

xy je poseben paket za risanje diagramov. Za njegovo uporabo moramo v preambulo dokumenta dodati:

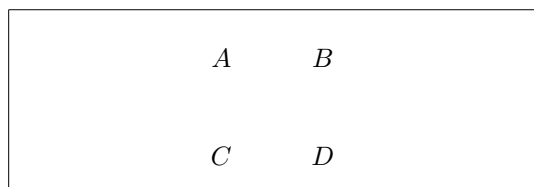
```
\usepackage[opcije]{xy}
```

Pri tem v *opcije* navedemo seznam funkcij iz Xy-pic ki jih želimo naložiti. Te opcije pridejo prav kadar je potrebno iskati napake v paketu, sicer pa

je najbolj priporočljivo na mestu *opcije* podati `all`, s čimer povzročimo, da \LaTeX naloži vse ukaze iz paketa \Xy-pic .

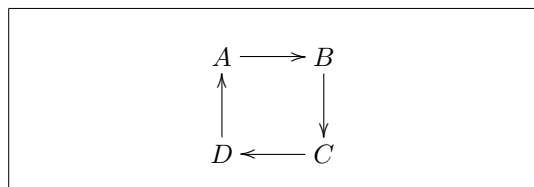
\Xy-pic diagrami se rišejo na podlago, ki je urejena v stilu matrike, pri čemer se vsak element diagrama zapiše na mesto elementa matrike:

```
\begin{displaymath}
\xymatrix{A & B \\
          C & D }
\end{displaymath}
```



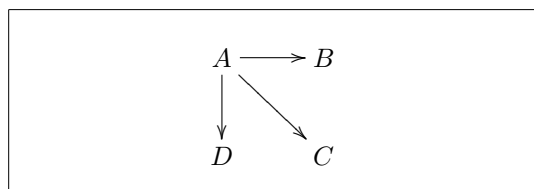
Ukaz `\xymatrix` moramo uporabiti v matematičnem načinu. V tem primeru smo podali dve vrstici in dva stolpca. Iz te matrike dobimo diagram, ko dodamo usmerjene povezave z ukazom `\ar`.

```
\begin{displaymath}
\xymatrix{ A \ar[r] & B \ar[d] \\
          D \ar[u] & C \ar[l] }
\end{displaymath}
```



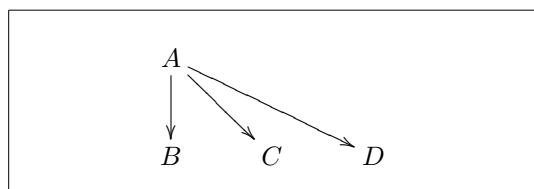
Ukaz za puščico je potrebno postaviti na mesto tistega elementa v matriki, od koder puščica izvira. Argumenti ukaza `ar` so smeri, kamor naj kaže puščica (u (gor), d (dol), r (desno) in l (levo)).

```
\begin{displaymath}
\xymatrix{
  A \ar[d] \ar[dr] \ar[r] & B \\
  D & C }
\end{displaymath}
```



Diagonalne puščice dobimo tako, da uporabimo več kot eno črko za opis smeri. Opis smeri lahko tudi sestavimo iz več črk in tako dobimo daljše diagonale.

```
\begin{displaymath}
\xymatrix{
  A \ar[d] \ar[dr] \ar[dr] & & \\
  B & C & D }
\end{displaymath}
```



Še bolj zanimive diagrame dobimo, če puščice opremimo z napisi. To dosežemo z uporabo običajnih operatorjev za indekse in potence..

```

\begin{displaymath}
\xymatrix{
A \ar[r]^f \ar[d]_g & B \\
D \ar[r]_{f'} & C }
\end{displaymath}

```

$$\begin{array}{ccc}
A & \xrightarrow{f} & B \\
g \downarrow & & \downarrow g' \\
D & \xrightarrow{f'} & C
\end{array}$$

Kot vidite, operatorje za napise na puščicah uporabljamo kot v matematičnem načinu. Edina razlika je, da sedaj potenca (zgornji indeks) pomeni »na vrhu puščice,« indeks pa pomeni »pod puščico.« Obstaja še tretji operator, pokončna črta: | Ta operator razporedi tekst *na* puščico.

```

\begin{displaymath}
\xymatrix{
A \ar[r]|f \ar[d]|g & B \\
D \ar[r]|{f'} & C }
\end{displaymath}

```

$$\begin{array}{ccc}
A & \xrightarrow{|f} & B \\
g \downarrow & & \downarrow g' \\
D & \xrightarrow{|{f'}} & C
\end{array}$$

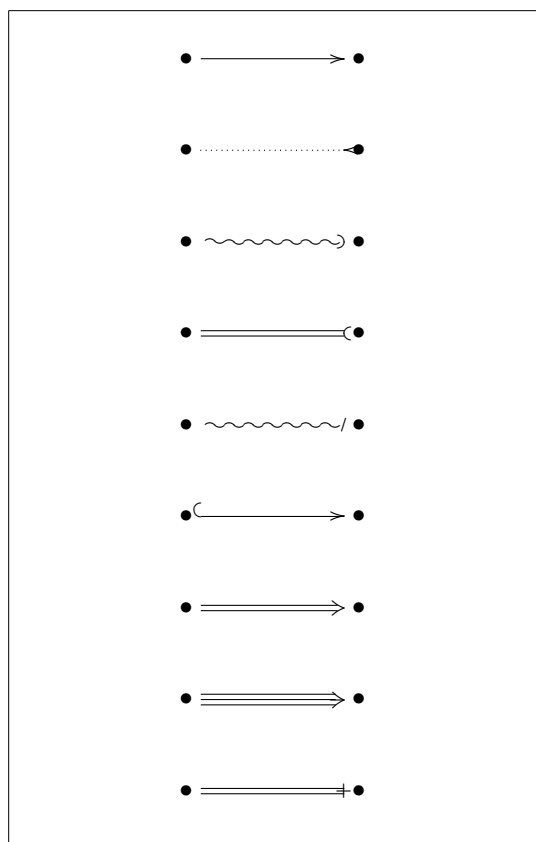
Če potrebujemo puščico z luknjo, uporabimo `\ar[...]|hole`.

V nekaterih primerih je potrebno ločiti med različnimi tipi puščic. To lahko naredimo tako, da jih opremimo z napisi ali pa spremenimo njihov videz:

```

\shorthandoff{"}
\begin{displaymath}
\xymatrix{
\bullet\ar@{->}[rr] && \bullet\\
\bullet\ar@{.<}[rr] && \bullet\\
\bullet\ar@{-~}[rr] && \bullet\\
\bullet\ar@{=()}[rr] && \bullet\\
\bullet\ar@{-/}[rr] && \bullet\\
\bullet\ar@{^{}(->}[rr] && \bullet\\
\bullet\ar@2{->}[rr] && \bullet\\
\bullet\ar@3{->}[rr] && \bullet\\
\bullet\ar@{=+}[rr] && \bullet
}
\end{displaymath}
\shorthandon{"}

```

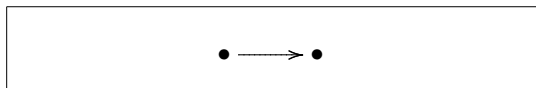


Bodite pozorni na razlike med naslednjima dvema diagramoma:

```

\begin{displaymath}
\xymatrix{
\bullet \ar[r]
\ar@{.>}[r] &
\bullet
}
\end{displaymath}

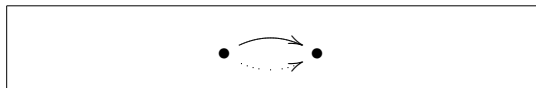
```



```

\begin{displaymath}
\xymatrix{
\bullet \ar/^/[r]
\ar@/_/@{.>}[r] &
\bullet
}
\end{displaymath}

```



Znamenje med poševnicama pove, kako naj se nariše krivulja. `Xy-pic` ponuja mnogo možnosti, ki vplivajo na risanje krivulj. Za več informacij pogledajte v dokumentacijo, ki je priložena paketu.

Poglavje 6

Prilagajanje L^AT_EXa

Dokumenti narejeni z ukazi, ki smo jih spoznali do tega trenutka, bodo sprejemljivi za večji del občinstva. Kljub temu da njihov videz ni razkošen, se držijo vseh uveljavljenih pravil dobrega stavljenja, kar jih naredi lahko berljive in prijetnega videza.

Toda, obstajajo situacije v katerih L^AT_EX nima na voljo ukaza ali okolja, ki ustreza našim potrebam, ali pa rezultat dobljen z obstoječimi ukazi ne izpolni naših želja.

V tem poglavju bomo dali nekaj namigov, kako lahko L^AT_EX naučimo novih trikov in kako lahko naredimo dokumente z drugačnim videzom od običajno privzetega.

6.1 Novi ukazi, okolja in paketi

Verjetno ste že opazili, da so vsi ukazi, ko so prvič predstavljeni v knjigi, uokvirjeni, hkrati pa se pojavijo tudi v stvarnem kazalu na koncu knjige. Namesto direktne uporabe potrebnih L^AT_EXovih ukazov, s katerimi se da to doseči, sem napisal paket, v katerem so definirani novi ukazi in okolja za ta namen. Tako lahko preprosto napišemo:

```
\begin{lscommand}  
\ci{dum}  
\end{lscommand}
```



V tem zgledu smo uporabili novo okolje `lscommand`, ki nariše okvir okrog ukaza in nov ukaz `\ci`, ki zapiše ime ukaza, hkrati pa ustrezeni podatek vstavi še v stvarno kazalo. To lahko preverite tako, da v stvarnem kazalu na koncu knjige poiščete geslo `\dum`. Tam boste pri `\dum` našli referenco na vsako stran, kjer je v tekstu ukaz `\dum`.

Če se odločimo, da nočemo več imeti novih ukazov v okvirjih, lahko preprosto popravimo definicijo okolja `lscommand` in naredimo nov videz. To

je veliko enostavneje kot pa v celem dokumentu popravljati ustrezen tekst na vseh mestih, kjer se direktno uporabljajo L^AT_EXovi ukazi za risanje okvirja okrog besede.

6.1.1 Novi ukazi

Svoje nove ukaze lahko definiramo z ukazom

```
\newcommand{ime}[num]{definicija}
```

Ukaz potrebuje dva argumenta: *ime* je ime ukaza, ki ga želimo definirati, *definicija* pa je opis tega, kar želimo, da se izvede, zapisan z ustreznimi L^AT_EX ukazi. Argument *num* v oglatih oklepajih je neobvezen in določa število argumentov, ki jih potrebuje naš novi ukaz (maksimalno možno število parametrov je 9). Če tega argumenta ni, se privzame vrednost 0, kar pomeni, da gre za ukaz brez argumentov.

Naslednja dva zgleda bosta zadevo še bolj razjasnila. V prvem zgledu definiramo nov ukaz z imenom `\xvec`. Ta ukaz nam pride prav vsakič ko je potrebno izpisati vektor x_1, \dots, x_n . Namesto z `\$x_1, \ldots, x_n\$` lahko sedaj to naredimo z `\$xvec\$`.

```
\newcommand{\xvec}{x_1, \ldots, x_n}
Vektor \$xvec$ \ldots
```

Vektor $x_1, \dots, x_n \dots$

Naslednji zgled prikazuje, kako definiramo ukaz, ki potrebuje en argument. Denimo, da v tekstu poleg x_1, \dots, x_n večkrat potrebujemo tudi y_1, \dots, y_n . Namesto definicije `\yvec` definiramo splošni ukaz z enim argumentom. Značka `#1` se zamenja z argumentom, ki ga podamo v zavutih oklepajih. Če definiramo ukaz z več kot enim argumentom, potem je naslednji označen z `#2` in tako naprej.

```
% v preambuli:
\newcommand{\lvec}[1]
  {#1_1, \ldots, #1_n}
% v telesu dokumenta:
Skalarni produkt vektorjev
\$lvec{x}$ in \$lvec{y}$ je
\ldots
```

Skalarni produkt vektorjev x_1, \dots, x_n in y_1, \dots, y_n je \dots

L^AT_EX ne dovoli da definiramo nov ukaz, ki bi povozil že obstoječega. Če to na vsak način želimo narediti, imamo na voljo ukaz `\renewcommand`. Način uporabe je povsem enak kot pri ukazu `\newcommand`.

V nekaterih primerih pride v poštev tudi ukaz `\providecommand`. Deluje podobno kot `\newcommand`, toda če ukaz že obstaja, potem L^AT_EX 2_ε tiho ignorira ukaz z definicijo in ohrani stari ukaz.

Tu se spleča še enkrat spomniti, kako je s presledki za L^AT_EX ukazi. Če se ne spomnite več, poglejte na stran 5.

6.1.2 Nova okolja

Podobno kot ukaz `\newcommand`, obstaja tudi ukaz, s katerim lahko definiramo nova okolja. Ukaz `\newenvironment` ima naslednjo obliko:

```
\newenvironment{ime}[num]{preden}{potem}
```

Podobno kot pri ukazu `\newcommand`, lahko `\newenvironment` uporabljamo z neobveznim argumentom ali pa brez njega. To, kar navedemo v argumentu *preden*, se izvede pred procesiranjem teksta v okolju, vsebina *potem* pa se procesira takrat, ko srečamo ukaz `\end{ime}`.

Spodnji primer prikazuje uporabo ukaza `\newenvironment`.

```
\newenvironment{kralj}
{\rule{1ex}{1ex}%
 \hspace{\stretch{1}}}
{\hspace{\stretch{1}}%
 \rule{1ex}{1ex}}
```

■ Moji skromni podložniki ... ■

```
\begin{kralj}
Moji skromni podložniki \ldots
\end{kralj}
```

Argument *num* ima podoben pomen kot pri ukazu `\newcommand`. L^AT_EX poskrbi za to, da ne moremo definirati okolja, ki že obstaja. Če želimo spremeniti kakšno že obstoječo okolje, uporabimo ukaz `\renewenvironment`, ki ima enak način uporabe kot ukaz `\newenvironment`.

Ukazi, uporabljeni v zgornjem primeru, bodo razloženi v nadaljevanju. Za ukaz `\rule` poglejte stran 115, za `\stretch` stran 108, več informacij o `\hspace` pa dobite na strani 108.

6.1.3 Presledki na začetku in po koncu okolja

Ko kreiramo novo okolje, se nam lahko zgodi, da se na začetku ali na koncu brez kakšnega vidnega vzroka pojavijo nezaželeni presledki. Denimo, da želimo definirati okolje za naslov, ki ne bo zamaknjen, prav tako pa ne bo zamaknjen prvi odstavek za naslovom. Ukaz `\ignorespaces` na začetku definicije okolja povzroči, da se ignorirajo vsi morebitni presledki za ukazom `\begin{okolje}`. Še bolj zapleteno je poskrbeti za dogajanje po koncu bloka, saj na koncu okolja pride do posebne obdelave. Z ukazom `\ignorespacesafterend` lahko L^AT_EXu naročimo, naj za ustreznim `\end{okolje}` ukazom, ko bo posebne obdelave konec, sproži še ukaz `\ignorespaces`.

```
\newenvironment{enostavno}%
{\noindent}%
{\par\noindent}

\begin{enostavno}
Pozor na presledek\\na levi strani.
\end{enostavno}
Enako\\tukaj.
```

Pozor na presledek na levi strani. Enako tukaj.

```
\newenvironment{pravilno}%
{\noindent\ignorespaces}%
{\par\noindent%
\ignorespacesafterend}

\begin{pravilno}
Ni presledka\\na levi strani.
\end{pravilno}
Enako\\tukaj.
```

Ni presledka na levi strani. Enako tukaj.

6.1.4 Poganjanje L^AT_EXa iz ukazne vrstice

Če delate na operacijskem sistemu Unix oz. podobnem, lahko za prevajanje L^AT_EX projektov uporabljate make datoteke. V tem primeru zna biti za vas zanimivo, da lahko z uporabo dodatnih parametrov pri klicu prevajalnika iz istega dokumenta dobite različne različice. Npr. če v dokument dodamo naslednjo strukturo:

```
\usepackage{ifthen}
\ifthenelse{\equal{\crnabelo}{true}}{
  % "črno bel" način; naredi nekaj..
}{
  % "barvni" način; naredi nekaj drugega..
}
```

lahko potem dokument prevajamo iz ukazne vrstice z:

```
latex '\newcommand{\crnabelo}{true}\input{test.tex}'
```

Na ta način se najprej definira ukaz `\blackandwhite`, potem pa se prevede še izbrana datoteka in v njej se prevedejo deli dokumenta v črno belem načinu. Če bi namesto tega uporabili

```
latex '\newcommand{\crnabelo}{false}\input{test.tex}'
```

bi dobili barvno verzijo dokumenta.

6.1.5 Lastni paketi

Če definiramo veliko novih okolij in ukazov, potem bo preambula dokumenta postala kar dolga. V takem primeru je dobro narediti nov \LaTeX ov paket, ki vsebuje vse definicije novih okolij in ukazov. V dokumentu potem uporabimo ukaz `\usepackage`, da naložimo paket in s tem v dokumentu omogočimo nove ukaze in okolja.

```
% Demo Package by Tobias Oetiker and Bor Plestenjak
\ProvidesPackage{demopack}
\newcommand{\xvec}[1] {x_1,\ldots,x_n}
\newcommand{\lvec}[1] {\#1_1,\ldots,\#1_n}
\newenvironment{kralj}
  {\rule{1ex}{1ex} \hspace{\stretch{1}}}
  {\hspace{\stretch{1}} \rule{1ex}{1ex}}
```

Slika 6.1: Zgled paketa.

Pisanje paketa v glavnem pomeni kopiranje vsebine preambule v ločeno datoteko s končnico `.sty`. Poleg tega je še poseben ukaz

`\ProvidesPackage{package name}`

ki ga je potrebno uporabiti povsem na začetku datoteke s paketom. Ukaz `\ProvidesPackage` sporoči \LaTeX u ime paketa in javi resno napako v primeru, ko želimo paket naložiti dvakrat. Slika 6.1 prikazuje majhen paket z ukazi, ki smo jih definirali v prejšnjih zgledih.

6.2 Pisave in velikosti črk

6.2.1 Ukazi za spreminjanje pisave

\LaTeX izbere pisavo in velikost črk glede na logično strukturo dokumenta (razdelki, opombe, ...). V določenih primerih pa bi radi ročno spremenili pisavo in velikosti črk. To lahko naredimo z ukazi, navedenimi v tabelah 6.1 in 6.2. Dejanska velikost vsake pisave je oblikovalski problem in je odvisna od razreda dokumenta in uporabljenih opcij. Tabela 6.3 prikazuje absolutno velikost črk pri uporabljenih navedenih ukazih v standardnih razredih dokumentov.

```
{\small The small and
\textbf{bold} Romans ruled}
{\Large all of great big
\textit{Italy}.}
```

The small and **bold** Romans ruled all of great big *Italy*.

Pomembna lastnost L^AT_EX_{2 ϵ} je, da so vsi atributi pisave neodvisni. To pomen, da lahko vključimo ukaze za spreminjanje velikosti ali tipa pisave, pa se bodo še vedno ohranili atributi za krepko ali poševno pisavo, ki smo jih vključili prej.

V *matematičnem načinu* lahko uporabljamo ukaze za spreminjanje pisave tako, da gremo začasno ven iz *matematičnega načina* in vnesemo normalen tekst. Če želimo pri stavljenju formul uporabljati drugačno pisavo, potem za to obstajajo posebno ukazi. Našteti so v tabeli 6.4.

V povezavi z ukazi za spreminjanje velikosti črk imajo velik pomen zaviti oklepaji. Z njimi gradimo *skupine*. Skupine omejujejo območje delovanja L^AT_EXovega ukaza.

He likes {\LARGE large and
\small small} letters.

He likes large and small letters.

Ukazi za spreminjanje velikosti pisave spremenijo tudi razmike med vrsticami, toda le, če se odstavek konča znotraj območja delovanja ukaza za velikost pisave. Desni zaviti oklepaj } zato ne sme nastopati prezgodaj v

Tabela 6.1: Pisave.

<code>\textrm{...}</code>	pokončna pisava	<code>\textsf{...}</code>	gladka pisava
<code>\texttt{...}</code>	pisalni stroj		
<code>\textmd{...}</code>	srednja pisava	<code>\textbf{...}</code>	krepka pisava
<code>\textup{...}</code>	pokončna pisava	<code>\textit{...}</code>	<i>kurzivna pisava</i>
<code>\textsl{...}</code>	<i>nagnjena pisava</i>	<code>\textsc{...}</code>	VELIKE MALE ČRKE
<code>\emph{...}</code>	<i>poudarjena pisava</i>	<code>\textnormal{...}</code>	običajna pisava font

Tabela 6.2: Velikosti črk.

<code>\tiny</code>	drobna pisava	<code>\Large</code>	veliki znaki
<code>\scriptsize</code>	velikost indeksov	<code>\LARGE</code>	zelo veliki znaki
<code>\footnotesize</code>	velikost opomb pod črto		
<code>\small</code>	majhna pisava	<code>\huge</code>	ogromni znaki
<code>\normalsize</code>	normalna velikost	<code>\Huge</code>	največji znaki
<code>\large</code>	veliki znaki		

Tabela 6.3: Absolutna velikost pisave v standardnih razredih.

size	10pt (privzeto)	11pt opcija	12pt opcija
<code>\tiny</code>	5pt	6pt	6pt
<code>\scriptsize</code>	7pt	8pt	8pt
<code>\footnotesize</code>	8pt	9pt	10pt
<code>\small</code>	9pt	10pt	11pt
<code>\normalsize</code>	10pt	11pt	12pt
<code>\large</code>	12pt	12pt	14pt
<code>\Large</code>	14pt	14pt	17pt
<code>\LARGE</code>	17pt	17pt	20pt
<code>\huge</code>	20pt	20pt	25pt
<code>\Huge</code>	25pt	25pt	25pt

Tabela 6.4: Matematične pisave.

<i>Ukaz</i>	<i>Zgled</i>	<i>Rezultat</i>
<code>\mathcal{...}</code>	<code>\$\$\mathcal{B}=c\$</code>	$\mathcal{B} = c$
<code>\mathrm{...}</code>	<code>\$\$\mathrm{K}_2\$</code>	K_2
<code>\mathbf{...}</code>	<code>\$\$\sum x=\mathbf{v}\$</code>	$\sum x = \mathbf{v}$
<code>\mathsf{...}</code>	<code>\$\$\mathsf{G}\times\mathsf{R}\$</code>	$G \times R$
<code>\mathhtt{...}</code>	<code>\$\$\mathhtt{L}(b,c)\$</code>	$L(b, c)$
<code>\mathnormal{...}</code>	<code>\$\$\mathnormal{R_{19}}\neq R_{19}\$</code>	$R_{19} \neq R_{19}$
<code>\mathit{...}</code>	<code>\$\$\mathit{ffi}\neq ffi\$</code>	$ffi \neq ffi$

tekstu. Primerjajte položaj ukaza `\par` v naslednjih dveh primerih.¹

```
{\Large Don't read this! It is not
true. You can believe me!}\par}
```

Don't read this! It is not true.
You can believe me!

```
{\Large This is not true either.
But remember I am a liar.}\par}
```

This is not true either. But
remember I am a liar.

Če želimo, da ukaz za spremenjeno velikost pisave deluje za celotni odstavek ali celo za večji kos teksta, potem je priporočljivo uporabljati okolja za spreminjanje velikosti pisave.

```
\begin{Large}
This is not true.
But then again, what is these
days \ldots
\end{Large}
```

This is not true. But then
again, what is these days ...

To nas lahko reši pred štejetjem velikega števila zavitih oklepajev.

6.2.2 Nevarnost na vidiku

Kot smo omenili že na začetku tega poglavja, je nevarno razmetati eksplisitne ukaze za spreminjanje oblike pisave vsepovsod po tekstu, saj je to v nasprotju z osnovno idejo L^AT_EXa, ki pravi, da je potrebno ločiti oznake za logični in vizualni del dokumenta. To pomeni, da če uporabljamo isti ukaz za spreminjanje pisave na več mestih z namenom, da poudarimo določen podatek, potem je bolje za to definirati nov ukaz preko `\newcommand` in tako logično povezati vrsto podatka, ki ga poudarjamo s spreminjanjem pisave.

```
\newcommand{\oops}[1]{\textbf{#1}}
Do not \oops{enter} this room,
it's occupied by a \oops{machine}
of unknown origin and purpose.
```

Do not **enter** this room, it's occupied by a
machine of unknown origin and purpose.

Ta pristop ima to prednost, da če se kdaj kasneje odločimo, da bomo za nevarnost uporabili drugačno pisavo kot pa `\textbf`, potem to spremenimo na enem mestu. Sicer bi morali v celotnem dokumentu poiskati vse pojavitve ukaza `\textbf`, potem pa bi se morali pri vsakem posamezno še odločiti, ali je ukaz `\textbf` uporabljen zaradi nevarnosti in moramo pisavo zato zamenjati ali pa ukaz `\textbf` pomeni kaj drugega in ga pustimo pri miru.

1. `\par` je ekvivalentno prazni vrstici

6.2.3 Nasvet

Za konec našega izleta v deželo pisav in velikosti črk še kratek nasvet:

Pomnite! Čim **VE** č pisav uporabljate v vašem dokumentu, tem lažje BERLJIV in *lepši postanete*.

6.3 Presledki

6.3.1 Razmik med vrsticami

Če želimo uporabljati v dokumentu večje razmike med vrsticami, potem lahko to spremenimo z ukazom

```
\linespread{factor}
```

v preambuli dokumenta. Vrednost `\linespread{1.3}` ustreza »ena in polovičnemu«, vrednost `\linespread{1.6}` pa »dvojnemu« razmiku med vrsticami. Normalno vrstice niso razmaknjene, zato je privzeta vrednost 1.

Bodite pozorni na to, da je rezultat, ki ga dobimo z uporabo ukaza `\linespread` dokaj drastičen in ni primeren za objavljena dela. Če imate dober razlog za spreminjanje privzetih razmikov med vrsticami, zato raje uporabljajte ukaz:

```
\setlength{\baselineskip}{1.5\baselineskip}
```

```
{\setlength{\baselineskip}%
  {1.5\baselineskip}
This paragraph is typeset with
the baseline skip set to 1.5 of
what it was before. Note the par
command at the end of the
paragraph. \par}
```

This paragraph has a clear purpose, it shows that after the curly brace has been closed, everything is back to normal.

This paragraph is typeset with the baseline skip set to 1.5 of what it was before. Note the par command at the end of the paragraph.

This paragraph has a clear purpose, it shows that after the curly brace has been closed, everything is back to normal.

6.3.2 Oblikovanje odstavka

V \LaTeX u imamo dva parametra, ki vplivata na obliko odstavka. Z vključitvijo definicije kot npr.

```
\setlength{\parindent}{0pt}
\setlength{\parskip}{1ex plus 0.5ex minus 0.2ex}
```

v preambulo dokumenta lahko spremenimo obliko odstavkov. Prvi ukaz nastavi zamik prve vrstice v odstavku na 0 (brez zamika), drugi ukaz pa nastavi navpični razmik med odstavki.

Vrednosti navedeni za `plus` in `minus` povesta T_EXu, za koliko lahko maksimalno skrči oziroma razširi predpisani razmik med odstavkoma, da se bodo odstavki lepo poravnali na stran.

V Evropi so odstavki pogosto ločeni z določenim razmikom, začetne vrstice pa niso zamaknjene. Toda pazite, saj to vpliva tudi na kazalo. Vrstice v kazalu so sedaj ločene z večjimi razmiki kot sicer. Da se izognemo temu, je bolje dva zgornja ukaza iz preambule prestaviti na neko mesto za `\tableofcontents` ali pa jih sploh ne uporabljati, saj večina knjig uporablja začetni zamik in ne navpični razmik za ločevanje odstavkov.

Če želimo zamakniti odstavek, ki ni zamaknjen, to naredimo z ukazom

```
\indent
```

na samem začetku odstavka.² To bo očitno imelo učinek le v primeru, ko vrednost `\parindent` ni nastavljena na 0.

Če želimo nezamaknjen odstavek, potem uporabimo

```
\noindent
```

kot prvi ukaz v odstavku. To pride v poštev, kadar začnemo besedilo kar s tekstom in ne z ukazom za definiranje poglavja, razdelka, ipd.

6.3.3 Vodoravni razmiki in zapolnjevalci

L^AT_EX avtomatično določi presledek med besedami in stavki. Če želimo dodati vodoravni razmik, uporabimo ukaz

```
\hspace{dolžina}
```

Če naj se ta presledek obdrži tudi v primeru, ko pade na začetek ali na konec vrstice, uporabimo `\hspace*` namesto `\hspace`. Argument *dolžina* je v enostavni obliki enak številu in merski enoti. Najpomembnejše merske enote so našteje v tabeli 6.5.

Ta `\hspace{1.5cm}`razmik ima dolžino 1.5 cm.

Ta razmik ima dolžino 1.5 cm.

2. Če želite zamakniti prvi odstavek v vsakem razdelku, potem uporabite paket `indentfirst` iz svežnja 'tools'.

Tabela 6.5: T_EX Merske enote.

mm	milimeter $\approx 1/25$ inch	▯
cm	centimeter = 10 mm	▯
in	palec (inč) = 25.4 mm	▯
pt	točka (pika) $\approx 1/72$ inča $\approx \frac{1}{3}$ mm	▯
em	približna širina 'M' v trenutni pisavi	▯
ex	približna višina 'x' v trenutni pisavi	▯

Ukaz

`\stretch{n}`

naredi poseben raztegljiv presledek. Razteza se dokler ne zapolni ves preostali prostor na vrstici. Če uporabimo dva ukaza `\hspace{\stretch{n}}` v isti vrstici, potem se bosta razširila glede na faktor širjenja.

`x\hspace{\stretch{1}}`
`x\hspace{\stretch{3}}x`

x x x

Ko uporabljamo vodoravne presledke med tekstom, je smiselno velikost presledkov relativno prilagoditi velikosti izbrane pisave. To lahko dosežemo z merskima enotama `em` in `ex`:

`{\Large}velik\hspace{1em}y\{\{\`
`{\tiny}drobni\hspace{1em}y}`

velik y
 drobni y

Ukaz

`\hfill`

je okrajšava za `\hspace{\fill}`. Tu je `\fill` posebna raztegljiva dolžina, ki se lahko od 0 raztegne do maksimalne možne širine. Ukaza

`\dotfill` in `\hrulefill`

delujeta tako kot `\hfill`, le da vmesni prostor zapolnita s pikami oziroma z vodoravno črto.

```
Začetek \dotfill\ Konec\
Levo\ \hrulefill\ Sredina\
\hrulefill\ Desno\
X\ \hfill\hfill Malo na desno
\ \hfill\ X
```

Začetek	Konec
Levo _____	Sredina _____ Desno
X	Malo na desno X

6.3.4 Navpični presledki

Razmik med odstavki, razdelki, podrazdelki, ... je v L^AT_EXu avtomatično določen. Kadar je potrebno, lahko dodatni navpični razmik *med dvema odstavkoma* vstavimo z ukazom:

```
\vspace{length}
```

Ta ukaz naj bi bil normalno uporabljen med dvema praznima vrsticama. Če želimo zadržati prostor na vrhu ali na dnu strani, potem lahko uporabljamo ukaz `\vspace*` namesto `\vspace`.

Ukaz `\stretch` v povezavi z ukazom `\pagebreak` lahko uporabimo za to, da tekst vstavimo na zadnjo vrstico strani ali pa da tekst navpično postavimo na sredo strani.

```
Nekaj teksta \ldots
```

```
\vspace{\stretch{1}}
```

```
To gre na zadnjo vrstico strani.\pagebreak
```

Dodatni razmik med dvema vrsticama v *istem* odstavku je določen z ukazom

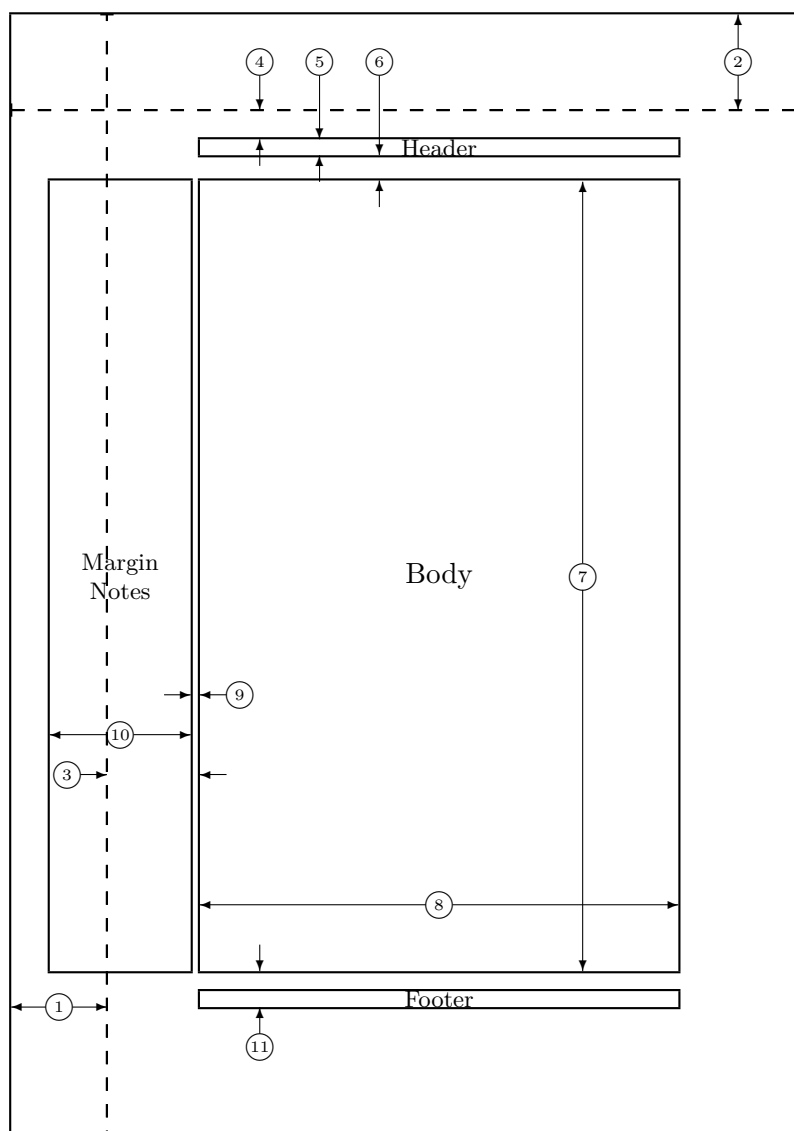
```
\[dolžina]
```

Z ukazom `\bigskip` in `\smallskip` lahko naredimo navpične razmike že vnaprej definiranih velikosti in nam tako ni potrebno skrbeti za točna števila.

6.4 Oblika strani

L^AT_EX 2_ε nam omogoča da v ukazu `\documentclass` podamo velikost strani. Velikost robov za tekst se potem določi avtomatično. V nekaterih primerih lahko nismo zadovoljni s privzetimi vrednostmi in te vrednosti se seveda da ročno spremeniti. Slika 6.2 prikazuje vse parametre, ki se jih da spremeniti. Narejena je bila s paketom `layout` iz svežnja 'tools'³.

3. CTAN:/tex-archive/macros/latex/required/tools



1	one inch + \hoffset	2	one inch + \voffset
3	\oddsidemargin = 22pt or \evensidemargin	4	\topmargin = 22pt
5	\headheight = 12pt	6	\headsep = 19pt
7	\textheight = 595pt	8	\textwidth = 360pt
9	\marginparsep = 7pt	10	\marginparwidth = 106pt \marginparpush = 5pt (not shown)
11	\footskip = 27pt \hoffset = 0pt \paperwidth = 597pt		\voffset = 0pt \paperheight = 845pt

Slika 6.2: Parametri oblike strani.

POČAKAJTE! ...preden začnete razmišljati v stilu »Naredimo to ozko stran malce širšo«, si vzemite nekaj sekund za razmislek. Kakor velja za večino stvari v L^AT_EXu, je tudi tu dober razlog za to, da je oblika strani takšna kot je

Seveda, če jo primerjamo s stranjo, narejeno s programom MS Word, zgleda grozno ozka. Toda poglejte v vašo priljubljeno knjigo⁴ in preštejte število znakov v povprečni vrstici teksta. Opazili boste, da v vrstici ni več kot 66 znakov. Isto ponovite na vaši L^AT_EXovi strani. Opazili boste, da je tudi tu okrog 66 znakov na stran. Izkušnje kažejo, da branje postane težje takoj, ko vrstica vsebuje več znakov. To je zaradi tega, ker je oči težko premikati od konca ene vrstice na začetek naslednje. To je med drugim tudi en izmed razlogov, da je besedilo v časopisu v več stolpcih.

Če torej povečate širino telesa s tekstom, mislite tudi na to, da s tem otežujete življenje vašim bralcem. Naj bo sedaj konec opozoril in si poglejmo, kako lahko vseeno spremenimo obliko strani.

Za spreminjanje parametrov ima L^AT_EX na voljo dva ukaza. Ponavadi jih uporabljamo v preambuli dokumenta.

Prvi ukaz določi fiksno vrednost izbranemu parametru:

```
\setlength{parameter}{dolžina}
```

Drugi ukaz doda dolžino izbranemu parametru:

```
\addtolength{parameter}{dožina}
```

Drugi ukaz je bolj praktičen od `\setlength`, saj lahko obliko spreminjamo relativno glede na obstoječo obliko. Če želimo širino teksta povečati za en centimeter, v preambulo dodamo naslednje ukaze:

```
\addtolength{\hoffset}{-0.5cm}
\addtolength{\textwidth}{1cm}
```

Tu se mogoče splača pogledati paket `calc`, ki nam omogoča da v argumentih ukaza `\setlength` in drugih, kjer vnašamo numerične vrednosti, uporabljamo aritmetične operacije.

6.5 Še več zabave z dolžinami

Kadarkoli se da, se izogibam uporabi absolutnih dolžin v L^AT_EXovih dokumentih. Raje uporabljam za osnovne mere širino ali višino elementov na strani. Za širino slike je to npr. `\textwidth`, če želimo, da slika zapolni celo stran.

4. Tu mislim na pravo natisnjeno knjigo, ki jo je izdala priznana založba.

Naslednji trije ukazi določajo širino, višino in globino tekstovnega niza.

```
\settoheight{lsccommand}{tekst}
\settodepth{lsccommand}{tekst}
\settowidth{lsccommand}{tekst}
```

Naslednji zgled prikazuje možno uporabo teh ukazov.

```
\flushleft
\newenvironment{vardesc}[1]{%
  \settowidth{\parindent}{#1:\ }
  \makebox[Opt][r]{#1:\ }{ }
}

\begin{displaymath}
a^2+b^2=c^2
\end{displaymath}

\begin{vardesc}{Where}$a$,
$b$ -- are adjunct to the right
angle of a right-angled triangle.

$c$ -- is the hypotenuse of
the triangle and feels lonely.

$d$ -- finally does not show up
here at all. Isn't that puzzling?
\end{vardesc}
```

$$a^2 + b^2 = c^2$$

Where: a , b – are adjunct to the right angle of a right-angled triangle.

c – is the hypotenuse of the triangle and feels lonely.

d – finally does not show up here at all. Isn't that puzzling?

6.6 Škatle

\LaTeX sestavlja svoje strani z zlaganjem škatel. Kot prvo je vsaka črka majhna škatla, ta škatla pa se zloži z ostalimi črkami v škatlo za besedo. Beseda se zloži z ostalimi besedami, toda tokrat s posebnim vmesnim vezivom, ki je raztegljiv, kar omogoča, da se vrsta besed tako skrči oziroma raztegne, da zapolnjuje eno vrstico na strani.

Priznam, da je to zelo poenostavljen pogled na to, kar se v resnici dogaja, toda bistvo je v tem, da \TeX res deluje s škatlami in vmesnim vezivom. Škatle so lahko ne samo črke, pač pa lahko v škatlo vstavimo praktično karkoli, vključno z drugimi škatlami. Vsako škatlo potem \LaTeX obravnava tako, kot da gre za posamezno črko.

V prejšnjih poglavjih smo se že srečali s škatlami, le omenjali jih nismo eksplicitno. Tako npr. okolje `tabular` in ukaz `\includegraphics` naredita škatlo. To pomeni, da lahko zlahka dve tabeli ali sliki postavimo drugo zraven druge. Poskrbeti moramo le za to, da njuna kombinacija ni širša od širine teksta.

Poljuben odstavek lahko vložimo v škatlo ali z ukazom

```
\parbox[položaj]{širina}{tekst}
```

ali pa v okolju

```
\begin{minipage}[položaj]{širina} tekst \end{minipage}
```

Vrednost parametra *položaj* je ena izmed črk *c*, *t* ali *b* ki pove, kako se škatla navpično poravna glede na osnovnico besedila v tekoči vrstici. Pri tem *t* pomeni poravnavo zgornjih robov, *c* sredinsko poravnavo in *b* poravnana spodnja robova. Z argumentom *širina* podamo širino škatle. Glavna razlika med `minipage` in `parbox` je, da znotraj `parbox` ne moremo uporabljati vseh ukazov in okolij, medtem ko je v okolju `minipage` možno skoraj vse.

Medtem, ko `\parbox` vstavi celoten odstavek tako, da prelamlja vrstice, obstaja tudi razred ukazov za škatle, ki delujejo le na vodoravno poravnanih objektih. Enega že poznamo. To je ukaz `\mbox`, ki vrsto škatel sestavi v eno, ukaz pa ponavadi uporabljamo zato, da preprečimo L^AT_EXu, da bi prelomil vrstico med dvema besedama. Ker lahko škatle vstavljamo v nove škatle, nam ti ukazi za vodoravno sestavljanje škatel omogočajo zelo veliko možnosti.

V ukazu

```
\makebox[širina][položaj]{tekst}
```

argument *širina* definira širino škatle, kot je vidna od zunaj.⁵ Poleg numeričnih vrednosti za dolžino lahko v argumentu *širina* uporabljamo tudi ukaze `\width`, `\height`, `\depth` in `\totalheight`. Njihove vrednosti so odvisne od dimenzij stavljenega besedila v parametru *tekst*. Parameter *položaj* ima za vrednost eno črko: *c* pomeni sredinsko poravnavo, *l* levo poravnavo, *r* desno poravnavo in *s* obojestransko poravnavo teksta znotraj škatle.

Ukaz `\framebox` deluje tako kot `\makebox`, le da dodatno nariše še okvir okrog teksta.

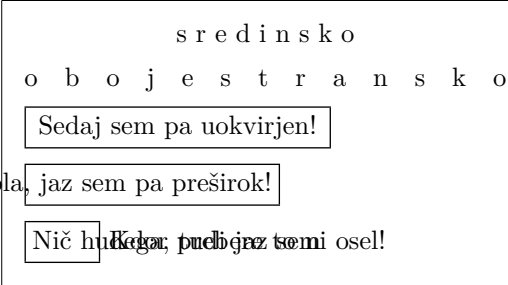
Naslednji zgledi prikazujejo, kaj vse se da narediti z ukazoma `\makebox` in `\framebox`.

5. To pomeni, da je škatla lahko manjša kot pa je širina objektov, ki jo sestavljajo. Širino lahko postavimo celo na 0pt, s čimer dosežemo, da se tekst izpiše, a nima nobenega vpliva na sosednje škatle.


```

\makebox[\textwidth]{%
  s r e d i n s k o}\par
\makebox[\textwidth][s]{%
  o b o j e s t r a n s k o}\par
\framebox[1.1\width]{Sedaj sem
  pa uokvirjen!} \par
\framebox[0.8\width][r]{Smola,
  jaz sem pa preširok!} \par
\framebox[1cm][l]{Nič hudega,
  tudi jaz sem}
Kdor prebere to ni osel!

```



Sedaj poznamo vodoravne škatle in naslednji korak je, da se lotimo navpičnih škatel.⁶ Tudi tu ni težav za L^AT_EX. Z ukazom

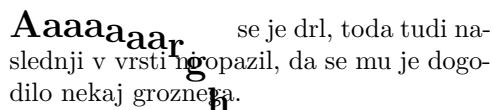
```
\raisebox{dvig}[globina][višina]{tekst}
```

lahko določimo navpične lastnosti škatle. V prvih treh parametrih lahko uporabljamo tudi `\width`, `\height`, `\depth` in `\totalheight`, da velikost določimo glede na velikost teksta v argumentu *tekst*.

```

\raisebox{0pt}[0pt][0pt]{\Large%
\textbf{Aaaa}\raisebox{-0.3ex}{a}%
\raisebox{-0.7ex}{aa}%
\raisebox{-1.2ex}[r]%
\raisebox{-2.2ex}{g}%
\raisebox{-4.5ex}[h]}
se je drl, toda tudi naslednji v
vrsti ni opazil, da se mu je
dogodilo nekaj groznega.

```



6.7 Poljubne črte

Nekaj strani nazaj ste mogoče opazili ukaz

```
\rule[dvig]{dolžina}{višina}
```

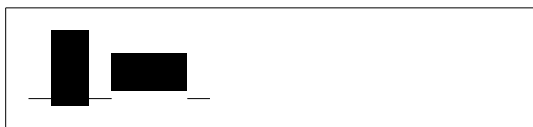
ki nariše črno črto z dano *dolžino* in debelino *višina* v višini *dvig* nad osnovnico tekoče vrstice. Parameter *dvig* je lahko tudi negativen.

6. Popoln nadzor lahko dobimo le s hkratno vodoravno in navpično kontrolo ...

```

\rule{3mm}{.1pt}%
\rule[-1mm]{5mm}{1cm}%
\rule{3mm}{.1pt}%
\rule[1mm]{1cm}{5mm}%
\rule{3mm}{.1pt}

```



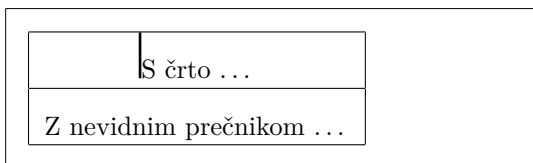
Ta ukaz uporabljamo za risanje navpičnih in vodoravnih črt. Črta na naslovnici je bila tako npr. narejena z ukazom `\rule`.

Poseben primer črte je takšna, ki nima širine, a ima določeno višino. V tiskarstvu se to imenuje prečnik. Uporabimo ga za to, da poskrbimo, da za določeni objekt na strani podamo minimalno višino. Uporabljamo ga lahko tudi v okolju `tabular`, s čimer poskrbimo, da imajo vse vrstice določeno minimalno višino.

```

\begin{tabular}{|c|}
\hline
\rule{1pt}{4ex}S črto \ldots\
\hline
\rule{0pt}{4ex}Z nevidnim
prečnikom \ldots \
\hline
\end{tabular}

```



Konec.

Literatura

- [1] Leslie Lamport. *L^AT_EX: A Document Preparation System*. Addison-Wesley, Reading, Massachusetts, second edition, 1994, ISBN 0-201-52983-1.
- [2] Donald E. Knuth. *The T_EXbook*, Volume A of *Computers and Typesetting*, Addison-Wesley, Reading, Massachusetts, second edition, 1984, ISBN 0-201-13448-9.
- [3] Frank Mittelbach, Michel Goossens, Johannes Braams, David Carlisle, Chris Rowley. *The L^AT_EX Companion*, (2nd Edition). Addison-Wesley, Reading, Massachusetts, 2004, ISBN 0-201-36299-6.
- [4] Michel Goossens, Sebastian Rahtz and Frank Mittelbach. *The L^AT_EX Graphics Companion*. Addison-Wesley, Reading, Massachusetts, 1997, ISBN 0-201-85469-4.
- [5] V vsaki namestitvi L^AT_EXa bi moral biti na voljo tudi t.i. *L^AT_EX Local Guide*, kjer so razložene zadeve, ki so vezane na posebnosti lokalnega sistema. Navodila bi morala biti v datoteki z imenom `local.tex`. Na žalost mnogo lenih sistemskih operaterjev ne pripravi teh navodil. V tem primeru za pomoč prosite lokalnega strokovnjaka za L^AT_EX.
- [6] L^AT_EX3 Project Team. *L^AT_EX 2_ε for authors*. Je vključeno v distribucijo L^AT_EX 2_ε v datoteki `usrguide.tex`.
- [7] L^AT_EX3 Project Team. *L^AT_EX 2_ε for Class and Package writers*. Je vključeno v distribucijo L^AT_EX 2_ε v datoteki `clsguide.tex`.
- [8] L^AT_EX3 Project Team. *L^AT_EX 2_ε Font selection*. Je vključeno v distribucijo L^AT_EX 2_ε v datoteki `fntguide.tex`.
- [9] D. P. Carlisle. *Packages in the 'graphics' bundle*. Je v svežnju 'graphics' v datoteki `grfguide.tex`, na voljo je na istem naslovu, s katerega ste dobili vašo distribucijo L^AT_EXa.
- [10] Rainer Schöpf, Bernd Raichle, Chris Rowley. *A New Implementation of L^AT_EX's verbatim Environments*. Je v svežnju 'tools' v datoteki

`verbatim.dtx`, na voljo je na istem naslovu, s katerega ste dobili vašo distribucijo \LaTeX a.

- [11] Vladimir Volovich, Werner Lemberg and \LaTeX 3 Project Team. *Cyrillic languages support in \LaTeX* . Je vključeno v distribucijo $\LaTeX 2_{\epsilon}$ v datoteki `cyrguide.tex`.
- [12] Graham Williams. *The TeX Catalogue*. To je zelo popoln seznam mnogih paketov za \TeX in \LaTeX . Na voljo je na naslovu [CTAN: /tex-archive/help/Catalogue/catalogue.html](http://ctan.org/tex-archive/help/Catalogue/catalogue.html)
- [13] Keith Reckdahl. *Using EPS Graphics in $\LaTeX 2_{\epsilon}$ Documents*. Tu je razloženo vse in več kot bi si kdaj želeli o EPS datotekah in njihovi uporabi v \LaTeX dokumentih. Na voljo je na naslovu [CTAN: /tex-archive/info/epslatex.ps](http://ctan.org/tex-archive/info/epslatex.ps)
- [14] Kristoffer H. Rose. *X_Y-pic User's Guide*. Na voljo je na CTAN z distribucijo X_Y-pic.
- [15] John D. Hobby. *A User's Manual for MetaPost*. Na voljo je na <http://cm.bell-labs.com/who/hobby/>
- [16] Alan Hoenig. *TeX Unbound*. Oxford University Press, 1998, ISBN 0-19-509685-1; 0-19-509686-X (pbk.)
- [17] Urs Oswald. *Graphics in $\LaTeX 2_{\epsilon}$* . Vsebuje nekaj programov v Javi za generiranje poljubnih krožnic in elips v okolju `picture` in navodila *MetaPost - A Tutorial*. Oboje je na voljo na <http://www.ursoswald.ch>
- [18] Vladimir Batagelj, Bojan Golli. *TeX Povabilo v TeX, LaTeX, BibTeX, PicTeX*. DMFA Slovenije, Ljubljana, 1990.
- [19] Aleš Košir. *Slovenščina in računalnik* je zelo podroben opis uporabe slovenščine v različnih operacijskih sistemih in programih, seveda tudi v \LaTeX u. Na voljo je na <http://nl.ijs.si/GNUs1/tex/tslovene/slolang/slolang.html>

Stvarno kazalo

- \!, 50
- ", 21
- \$, 43
- \(, 43
- \), 43
- \,, 44, 50
- , 22
- , 22
- \-, 20
- , 22
- , 22
- ., presledek za, 29
- ..., 23
- .aux, 14
- .cls, 13
- .dtx, 13
- .dvi, 13
- .fd, 13
- .idx, 14
- .ilg, 14
- .ind, 14
- .ins, 13
- .lof, 13
- .log, 13
- .lot, 13
- .sty, 13
- .tex, 11
- .toc, 13
- \:, 50
- \;, 50
- \@, 29
- \[, 44
- šumniki, 25
- \[, 19, 33, 34, 36, 110
- *, 19
- \], 44
- ~, 29
- abstract, 35
- \addtolength, 112
- Adobe Reader, 73
- æ, 25
- aeguill, 74
- akcenti, 24
- ambsy, 56
- amfonts, 45, 64
- amsmath, 44, 48–50, 52, 56
- amssymb, 45, 57
- amsthm, 54, 55
- \and, 30
- \appendix, 29, 31
- \ar, 96
- \arccos, 47
- \arcsin, 47
- \arctan, 47
- \arg, 47
- array, 50, 51
- \atop, 48
- \author, 30, 78

- babel, 20, 24, 25
- \backmatter, 31
- \backslash, 5
- barvni tekst, 11
- beamer, 81, 82
- \begin, 33, 86, 95
- \bibitem, 67
- \Big, 49
- \big, 49
- \Bigg, 49
- \bigg, 49
- \bigskip, 110

- `\binom`, 48
 - block, 83
 - bm, 56
- `\bmod`, 48
- `\boldmath`, 56
- `\boldsymbol`, 56
- calc, 112
- `\caption`, 39, 40
- `\cdot`, 47
- `\cdots`, 49
 - center, 33
- `\chapter`, 29
- `\chaptermark`, 70
- `\choose`, 48
- `\ci`, 99
- `\circle`, 89
- `\circle*`, 89
- `\cite`, 67
- `\cleardoublepage`, 40
- `\clearpage`, 40
- `\cline`, 36
 - color, 81
 - comment, 6
- `\cos`, 47
- `\cosh`, 47
- `\cot`, 47
- `\coth`, 47
- `\csc`, 47
- `\date`, 30
 - dcolumn, 37
- `\ddots`, 49
- `\deg`, 47
- `\depth`, 114, 115
 - description, 33
 - desna poravnava, 33
- `\det`, 47
 - diagonalne pike, 49
- `\dim`, 47
 - displaymath, 44
- `\displaystyle`, 53
 - dno strani, 11
 - doc, 12
- `\documentclass`, 9, 13, 20
 - dolžina, 108
 - dolge enačbe, 52
- `\dotfill`, 109
- `\dum`, 99
 - dva stolpca, 10
 - dvodelni pomišljaj, 22
 - dvojni razmik med vrsticami, 107
 - dvostranski izpis, 10
- eePIC, 85, 89
- `\emph`, 32, 104
 - empty, 11
 - en stolpec, 10
- Encapsulated PostScript, 65
- Encapsulated POSTSCRIPT, 75
- encodings
 - font
 - LGR, 27
 - OT1, 27
 - T1, 27
 - T2A, 27
 - T2B, 27
 - T2C, 27
 - X2, 27
 - input
 - utf8x, 26
- `\end`, 33, 86
 - enostranski izpis, 10
 - enumerate, 33
 - epic, 85
 - eqnarray, 51
- `\eqref`, 44
 - equation, 44
 - eurosym, 23
- `\EURtm`, 23
- `\exp`, 47
 - exscale, 12, 49
- fancyhdr, 69, 70
- `\fbox`, 21
 - figure, 38, 39
 - flushleft, 33
 - flushright, 33

- foiltex, 9
- \foldera, 93
- \folderb, 93
- font encodings, 27
 - LGR, 27
 - OT1, 27
 - T1, 27
 - T2A, 27
 - T2B, 27
 - T2C, 27
 - X2, 27
- fontenc, 12, 27
- \footnote, 31, 40
- \footnotesize, 104
- \footskip, 111
- formule, 43
- \frac, 48
- frame, 83
- \framebox, 114
- \frenchspacing, 29
- \frontmatter, 30
- \fussy, 20

- \gcd, 47
- \genfrac, 48
- geometry, 71
- geslo, 69
- gladka pisava, 104
- glava strani, 11
- gola i in j, 25
- grške črke, 46
- graphicx, 65, 75, 81

- \headheight, 111
- textttheadings, 11
- \headsep, 111
- \height, 114, 115
- \hfill, 109
- hipertekst, 72
- \hline, 36
- \hom, 47
- \href, 78, 80
- \hrulefill, 109
- \hspace, 101, 108

- \Huge, 104
- \huge, 104
- Hyperref, 79
- hyperref, 73, 76, 79–81
- hyphenat, 71
- \hyphenation, 20

- \idotsint, 50
- ifpdf, 80
- \ifpdf, 80
- ifthen, 12
- \ignorespaces, 101
- \ignorespacesafterend, 101
- \iiiint, 50
- \iiint, 50
- \iint, 50
- \include, 14
- \includegraphics, 66, 75, 80, 113
- \includeonly, 14
- indeks, 46
- \indent, 108
- indentfirst, 108
- \index, 68, 69
- \inf, 47
- \input, 14
- input encodings
 - utf8x, 26
- inputenc, 12, 26
- \int, 48
- integral, 48
- \item, 33
- itemize, 33

- jezik, 24

- \ker, 47
- Knuth, Donald E., 1
- kodna tabela, 12
- komentarji, 6
- končnica
 - .tex, 8
- končnice, 11
- koren, 46
- krepka pisava, 104
- krepki simboli, 56

- krhki ukazi, 40
- kurzivna pisava, 104
- `\label`, 31, 44
 - Lamport, Leslie, 1
- `\LARGE`, 104
- `\Large`, 104
- `\large`, 104
- `\LaTeX`, 21
 - L^AT_EX3, 4
- `\LaTeXe`, 21
 - l^at_exsym, 12
 - layout, 110
- `\ldots`, 23, 49
- `\left`, 49
- `\lefteqn`, 52
- `\leftmark`, 70
 - legal papir, 10
 - leva poravnava, 33
- `\lg`, 47
 - LGR, 27
 - ligature, 24
- `\lim`, 47
- `\liminf`, 47
- `\limsup`, 47
- `\line`, 88, 93
- `\linebreak`, 19
- `\linespread`, 107
- `\linethickness`, 90, 91, 93
- `\listoffigures`, 39
- `\listoftables`, 39
- `\ln`, 47
- `\log`, 47
 - longtable, 38
 - lsc_{command}, 99
- `\mainmatter`, 31, 79
- `\makebox`, 114
 - makeidx, 12, 68
 - makeidx paket, 68
- `\makeindex`, 68
 - makeindex program, 68
- `\maketitle`, 30
- `\marginparpush`, 111
- `\marginparsep`, 111
- `\marginparwidth`, 111
 - marvosym, 23
- matematične
 - funkcije, 47
- matematični
 - akcenti, 47
- matematični izrazi, 43
- matematični minus, 22
- matematični presledki, 50
- math, 43
- `\mathbb`, 45
- `\mathbf`, 105
- `\mathcal`, 105
- `\mathit`, 105
- `\mathnormal`, 105
- `\mathrm`, 53, 105
 - mathrsfs, 64
- `\mathsf`, 105
- `\mathtt`, 105
- `\max`, 47
- `\mbox`, 21, 24, 114
 - mednarodne nastavitve, 24
 - merska enota, 108
 - merske enote, 109
 - METAPOST, 76
- `\min`, 47
- minipage, 114
- minus, 22
- Mittelbach, Frank, 1
- mltex, 74
- mltex, 74
- modulska funkcija, 48
- `\multicolumn`, 37
- `\multipt`, 87, 90, 91
- nagnjena pisava, 104
- narekovaji, 21
- naslov, 10, 30
- naslov dokumenta, 10
- navpične pike, 49
- navpični presledek, 110
- `\newcommand`, 100
- `\newenvironment`, 101

- `\newline`, 19
- `\newpage`, 19
- `\newsavebox`, 92
- `\newtheorem`, 54
- `\newtheoremstyle`, 54
- `\noindent`, 108
- `\nolinebreak`, 19
- `\nonumber`, 52
- `\nopagebreak`, 19
- `\normalsize`, 104
- oblika strani, 110
 - empty, 11
 - headings, 11
 - plain, 11
- oblike strani, 11
- `\oddsidemargin`, 111
- odstavek, 17
- odvod, 47
- œ, 25
- oglati oklepaji, 5
- oklepaji, 49
- okolja, 32
- okolje
 - abstract, 35
 - array, 50, 51
 - block, 83
 - center, 33
 - comment, 6
 - description, 33
 - displaymath, 44
 - enumerate, 33
 - eqnarray, 51
 - equation, 44
 - figure, 38, 39
 - flushleft, 33
 - flushright, 33
 - frame, 83
 - itemize, 33
 - lscommand, 99
 - math, 43
 - minipage, 114
 - parbox, 114
 - picture, 85, 86, 89, 90
 - proof, 55
 - pspicture, 85
 - quotation, 34
 - quote, 34
 - subarray, 48
 - table, 38, 39
 - tabular, 36, 113
 - thebibliography, 67
 - verbatim, 35, 71
 - verse, 34
- opcije, 9
- opcijski parametri, 5
- osnovna velikost črk, 10
- OT1, 27
- `\oval`, 91, 93
- `\overbrace`, 46
- overflow hbox, 20
- `\overleftarrow`, 47
- `\overline`, 46
- `\overrightarrow`, 47
- `\pagebreak`, 19
- `\pageref`, 31, 72
- `\pagestyle`, 11
- paket, 6, 99
 - aeguill, 74
 - amsbsy, 56
 - amsfonts, 45, 64
 - amsmath, 44, 48–50, 52, 56
 - amssymb, 45, 57
 - amsthm, 54, 55
 - babel, 20, 24, 25
 - beamer, 81, 82
 - bm, 56
 - calc, 112
 - color, 81
 - dcolum, 37
 - doc, 12
 - eepic, 85, 89
 - epic, 85
 - eurosym, 23
 - exscale, 12, 49
 - fancyhdr, 69, 70
 - fontenc, 12, 27

- geometry, 71
- graphicx, 65, 75, 81
- Hyperref, 79
- hyperref, 73, 76, 79–81
- hyphenat, 71
- ifpdf, 80
- ifthen, 12
- indentfirst, 108
- inputenc, 12, 26
- latexsym, 12
- layout, 110
- longtable, 38
- makeidx, 12, 68
- marvosym, 23
- mathrsfs, 64
- mltex, 74
- ppower4, 81
- prosper, 81
- pstricks, 85, 89
- pxfonts, 75
- showidx, 69
- syntonly, 12, 14
- textcomp, 22
- txfonts, 75
- ucs, 26
- verbatim, 6, 71
- xy, 95
- paketi, 11
- \paperheight, 111
- \paperwidth, 111
- \par, 106
- \paragraph, 29
- parameter, 5
- \parbox, 114
- parbox, 114
- \parindent, 108
- \parskip, 108
- \part, 29
- PDF, 72
- pdfL^AT_EX, 73, 81
- pdfL^AT_EX, 73
- pdfT_EX, 73
- \phantom, 40, 52
- picture, 85, 86, 89, 90
- pika, 23
- pisava, 103
- plain, 11
- plavajoči objekti, 38
- \pmod, 48
- pogojni stavek, 12
- pokončna pisava, 104
- pomišljaj, 22
- poravnava na decimalni piki, 37
- posebni znaki, 24
- PostScript, 65
- POSTSCRIPT, 73, 74, 85, 86
 - Encapsulated, 75
- potenca, 46
- ppower4, 81
- \Pr, 47
- prečnik, 116
- preambula, 6
- preglas, 25
- prelom vrstice, 19
- presledek, 4
 - na začetku vrstice, 4
 - za ukazom, 5
- presledki, 4
- \printindex, 69
- \prod, 48
- produkt, 48
- proof, 55
- prosper, 81
- \protect, 40
- \providecommand, 100
- \ProvidesPackage, 103
- pspicture, 85
- pstricks, 85, 89
- puščica, 47
- \put, 87–92
- pxfonts, 75
- \qbezier, 85, 87, 94
- \qedhere, 55
- \qqquad, 44, 50
- \quad, 44, 50
- quotation, 34
- quote, 34

- `\raisebox`, 115
- razmik med odstavki, 108
- razmik med vrsticami, 107
- razred
 - article, 9
 - book, 9
 - letter, 9
 - proc, 9
 - report, 9
 - slides, 9
- `\ref`, 31, 44, 72
- `\renewcommand`, 100
- `\renewenvironment`, 101
- rezervirani znaki, 4
- `\right`, 49
- `\right.`, 49
- `\rightmark`, 70
- robovi, 110
- `\rule`, 101, 115, 116

- `\savebox`, 92
- `\scriptscriptstyle`, 53
- `\scriptsize`, 104
- `\scriptstyle`, 53
- `\sec`, 47
- `\section`, 29, 40
- `\sectionmark`, 70
- `\selectlanguage`, 26
- `\setlength`, 86, 108, 112
- `\settodepth`, 113
- `\settoheight`, 113
- `\settowidth`, 113
- seznam literature, 67
- showidx, 69
- `\sin`, 47
- `\sinh`, 47
- sistemi enačb, 51
- skandinavski znaki, 25
- sklicevanja, 31
- slike, 11, 65
- `\sloppy`, 20
- slovenščina, 27
- `\small`, 104
- `\smallskip`, 110

- `\sqrt`, 46
- `\stackrel`, 48
- `\stretch`, 101, 109
- struktura, 6
- stvarno kazalo, 68
- subarray, 48
- `\subparagraph`, 29
- `\subsection`, 29
- `\subsectionmark`, 70
- `\substack`, 48
- `\subsubsection`, 29
- `\sum`, 48
- `\sup`, 47
- syntonly, 12, 14

- T1, 27
- T2A, 27
- T2B, 27
- T2C, 27
- tabele, 36
- table, 38, 39
- `\tableofcontents`, 30
- tabular, 36, 113
- `\tan`, 47
- `\tanh`, 47
- telo strani, 11
- `\TeX`, 21
- `\texorpdfstring`, 79
- `\textbf`, 104
- `\textcelsius`, 22
- textcomp, 22
- `\texteuro`, 23
- `\textheight`, 111
- `\textit`, 104
- `\textmd`, 104
- `\textnormal`, 104
- `\textrm`, 53, 104
- `\textsc`, 104
- `\textsf`, 104
- `\textsl`, 104
- `\textstyle`, 53
- `\texttt`, 104
- `\textup`, 104
- `\textwidth`, 111

- thebibliography, 67
- \thicklines, 88, 91–93
- \thinlines, 91–93
- \thispagestyle, 11
- tilda (~), 22, 29
- \tiny, 104
- \title, 30
- \today, 21
- \topmargin, 111
- \totalheight, 114, 115
- tri pike, 23, 49
- txfonts, 75
- ucs, 26
- ukaz
 - !\, 50
 - \(, 43
 - \), 43
 - \,, 44, 50
 - \-, 20
 - \:, 50
 - \;, 50
 - \@, 29
 - \[, 44
 - \\, 19, 33, 34, 36, 110
 - *, 19
 - \], 44
 - \addtolength, 112
 - \and, 30
 - \appendix, 29, 31
 - \ar, 96
 - \arccos, 47
 - \arcsin, 47
 - \arctan, 47
 - \arg, 47
 - \atop, 48
 - \author, 30, 78
 - \backmatter, 31
 - \backslash, 5
 - \begin, 33, 86, 95
 - \bibitem, 67
 - \Big, 49
 - \big, 49
 - \Bigg, 49
 - \bigg, 49
 - \bigskip, 110
 - \binom, 48
 - \bmod, 48
 - \boldmath, 56
 - \boldsymbol, 56
 - \caption, 39, 40
 - \cdot, 47
 - \cdots, 49
 - \chapter, 29
 - \chaptermark, 70
 - \choose, 48
 - \ci, 99
 - \circle, 89
 - \circle*, 89
 - \cite, 67
 - \cleardoublepage, 40
 - \clearpage, 40
 - \cline, 36
 - \cos, 47
 - \cosh, 47
 - \cot, 47
 - \coth, 47
 - \csc, 47
 - \date, 30
 - \ddots, 49
 - \deg, 47
 - \depth, 114, 115
 - \det, 47
 - \dim, 47
 - \displaystyle, 53
 - \documentclass, 9, 13, 20
 - \dotfill, 109
 - \dum, 99
 - \emph, 32, 104
 - \end, 33, 86
 - \eqref, 44
 - \EURtm, 23
 - \exp, 47
 - \fbox, 21
 - \foldera, 93
 - \folderb, 93
 - \footnote, 31, 40
 - \footnotesize, 104

`\footskip`, 111
`\frac`, 48
`\framebox`, 114
`\frenchspacing`, 29
`\frontmatter`, 30
`\fussy`, 20
`\gcd`, 47
`\genfrac`, 48
`\headheight`, 111
`\headsep`, 111
`\height`, 114, 115
`\hfill`, 109
`\hline`, 36
`\hom`, 47
`\href`, 78, 80
`\hrulefill`, 109
`\hspace`, 101, 108
`\Huge`, 104
`\huge`, 104
`\hyphenation`, 20
`\idotsint`, 50
`\ifpdf`, 80
`\ignorespaces`, 101
`\ignorespacesafterend`, 101
`\iiiint`, 50
`\iiint`, 50
`\iint`, 50
`\include`, 14
`\includegraphics`, 66, 75, 80, 113
`\includeonly`, 14
`\indent`, 108
`\index`, 68, 69
`\inf`, 47
`\input`, 14
`\int`, 48
`\item`, 33
`\ker`, 47
`\label`, 31, 44
`\LARGE`, 104
`\Large`, 104
`\large`, 104
`\LaTeX`, 21
`\LaTeXe`, 21
`\ldots`, 23, 49
`\left`, 49
`\lefteqn`, 52
`\leftmark`, 70
`\lg`, 47
`\lim`, 47
`\liminf`, 47
`\limsup`, 47
`\line`, 88, 93
`\linebreak`, 19
`\linespread`, 107
`\linethickness`, 90, 91, 93
`\listoffigures`, 39
`\listoftables`, 39
`\ln`, 47
`\log`, 47
`\mainmatter`, 31, 79
`\makebox`, 114
`\makeindex`, 68
`\maketitle`, 30
`\marginparpush`, 111
`\marginparsep`, 111
`\marginparwidth`, 111
`\mathbb`, 45
`\mathbf`, 105
`\mathcal`, 105
`\mathit`, 105
`\mathnormal`, 105
`\mathrm`, 53, 105
`\mathsf`, 105
`\mathtt`, 105
`\max`, 47
`\mbox`, 21, 24, 114
`\min`, 47
`\multicolumn`, 37
`\multirow`, 87, 90, 91
`\newcommand`, 100
`\newenvironment`, 101
`\newline`, 19
`\newpage`, 19
`\newsavebox`, 92
`\newtheorem`, 54
`\newtheoremstyle`, 54
`\noindent`, 108

- `\nolinebreak`, 19
- `\nonumber`, 52
- `\nopagebreak`, 19
- `\normalsize`, 104
- `\oddsidemargin`, 111
- `\oval`, 91, 93
- `\overbrace`, 46
- `\overleftarrow`, 47
- `\overline`, 46
- `\overrightarrow`, 47
- `\pagebreak`, 19
- `\pageref`, 31, 72
- `\pagestyle`, 11
- `\paperheight`, 111
- `\paperwidth`, 111
- `\par`, 106
- `\paragraph`, 29
- `\parbox`, 114
- `\parindent`, 108
- `\parskip`, 108
- `\part`, 29
- `\phantom`, 40, 52
- `\pmod`, 48
- `\Pr`, 47
- `\printindex`, 69
- `\prod`, 48
- `\protect`, 40
- `\providecommand`, 100
- `\ProvidesPackage`, 103
- `\put`, 87–92
- `\qbezier`, 85, 87, 94
- `\qedhere`, 55
- `\qqquad`, 44, 50
- `\quad`, 44, 50
- `\raisebox`, 115
- `\ref`, 31, 44, 72
- `\renewcommand`, 100
- `\renewenvironment`, 101
- `\right`, 49
- `\right.`, 49
- `\rightmark`, 70
- `\rule`, 101, 115, 116
- `\savebox`, 92
- `\scriptscriptstyle`, 53
- `\scriptsize`, 104
- `\scriptstyle`, 53
- `\sec`, 47
- `\section`, 29, 40
- `\sectionmark`, 70
- `\selectlanguage`, 26
- `\setlength`, 86, 108, 112
- `\settodepth`, 113
- `\settoheight`, 113
- `\settowidth`, 113
- `\sin`, 47
- `\sinh`, 47
- `\sloppy`, 20
- `\small`, 104
- `\smallskip`, 110
- `\sqrt`, 46
- `\stackrel`, 48
- `\stretch`, 101, 109
- `\subparagraph`, 29
- `\subsection`, 29
- `\subsectionmark`, 70
- `\substack`, 48
- `\subsubsection`, 29
- `\sum`, 48
- `\sup`, 47
- `\tableofcontents`, 30
- `\tan`, 47
- `\tanh`, 47
- `\TeX`, 21
- `\texorpdfstring`, 79
- `\textbf`, 104
- `\textcelsius`, 22
- `\texteuro`, 23
- `\textheight`, 111
- `\textit`, 104
- `\textmd`, 104
- `\textnormal`, 104
- `\textrm`, 53, 104
- `\textsc`, 104
- `\textsf`, 104
- `\textsl`, 104
- `\textstyle`, 53
- `\texttt`, 104
- `\textup`, 104

- `\textwidth`, 111
- `\thicklines`, 88, 91–93
- `\thinlines`, 91–93
- `\thispagestyle`, 11
- `\tiny`, 104
- `\title`, 30
- `\today`, 21
- `\topmargin`, 111
- `\totalheight`, 114, 115
- `\underbrace`, 46
- `\underline`, 32, 46
- `\unitlength`, 86, 88
- `\usebox`, 92
- `\usepackage`, 11, 13, 23, 25–27, 103
- `\vdots`, 49
- `\vec`, 47
- `\vector`, 88
- `\verb`, 35
- `\verbatiminput`, 71
- `\vspace`, 110
- `\widehat`, 47
- `\widetilde`, 47
- `\width`, 114, 115
- `\xvec`, 100
- `\xymatrix`, 96
- ukazi, 5
- `\underbrace`, 46
- underfull hbox, 20
- `\underline`, 32, 46
- `\unitlength`, 86, 88
- URL, 22
- `\usebox`, 92
- `\usepackage`, 11, 13, 23, 25–27, 103
- utf8x, 26
- `\vdots`, 49
- `\vec`, 47
- `\vector`, 88
- vejica, 23
- vektor, 47
- velike male črke, 104
- velikost
 - oklepaj, 49
 - velikost črk, 104
 - velikost črk v dokumentu, 10
 - velikost A4, 10
 - velikost A5, 10
 - velikost B5, 10
 - velikost executive, 10
 - velikost letter, 10
 - velikost matematične pisave, 53
 - velikost pisave, 103
 - velikost strani, 10, 73, 110
- `\verb`, 35
- verbatim, 6, 71
- verbatim, 35, 71
- `\verbatiminput`, 71
- verse, 34
- vertikalne
 - pike, 49
- vezaj, 22
- vhodna datoteka, 7
- vodoravna
 - črta, 46
- vodoravne
 - pike, 49
- vodoravni
 - presledek, 108
 - zaviti oklepaj, 46
- vrste datotek, 11
- vsota, 48
- `\vspace`, 110
- `\widehat`, 47
- `\widetilde`, 47
- `\width`, 114, 115
- www, 22
- WYSIWYG, 2, 3
- X2, 27
- Xpdf, 73
- `\xvec`, 100
- xy, 95
- `\xymatrix`, 96
- zamik odstavka, 108
- zaviti oklepaji, 5, 104
- združevanje, 104

