

Nie príliš stručný úvod do systému L^AT_EX 2_ε

Alebo E^TE_X 2_ε za 95 minút

Tobias Oetiker

Hubert Partl, Irene Hyna a Elisabeth Schlegl

verzia 3.13, 23. február 2000

Preklad **Ján Buša ml. a st.**

posledná úprava 2. mája 2001

Copyright © 1999 Tobias Oetiker and all the Contributors to LShort. 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 program; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

Translation © 2001 Ján Buša ml. a st.

Vďaka!

Väčšina materiálu, použitého v tejto príručke, pochádza z rakúskeho *Úvodu do L^AT_EXu 2.09*, napísaného po nemecky týmito autormi:

Hubert Partl <partl@mail.boku.ac.at>

Zentraler Informatikdienst der Universität für Bodenkultur Wien

Irene Hyna <Irene.Hyna@bmf.ac.at>

Bundesministerium für Wissenschaft und Forschung Wien

Elisabeth Schlegl <no email>

in Graz

Ak máte záujem o nemecký variant tohoto manuálu, môžete jeho najnovšiu verziu, napísanú Jörgom Knappenom, nájsť na adrese:

CTAN:/tex-archive/info/lkurz

Počas prípravy tohoto dokumentu som požiadal o zaslanie recenzie na `comp.text.tex`. Dostal som množstvo ohlasov. Nasledujúci jednotlivci mi pomohli opravami, návrhmi a poskytli mi materiál na zlepšenie tejto práce. Venovali veľa úsilia, aby mi pomohli dostať tento dokument do jeho terajšej podoby. Rád by som im úprimne poďakoval. Prirodzene, všetky chyby, ktoré nájdete v tejto knižke, sú moje. Ak nájdete slovo, ktoré je vyslabikované správne, musel ma na to upozorniť písomne niekto z nižšie uvedených ľudí.

Rosemary Bailey, Friedemann Brauer, David Carlisle, Christopher Chin, Chris McCormack, Wim van Dam, Michael John Downes, David Dureisseix, Elliot, David Frey, Robin Fairbairns, Erik Frisk, Frank, Alexandre Guimond, Cyril Goutte, Greg Gamble, Neil Hammond, Rasmus Borup Hansen, Martien Hulsen, Werner Icking, Jakob, Eric Jacoboni, Alan Jeffrey, Byron Jones, David Jones, Johannes-Maria Kaltenbach, Andrzej Kawalec, Alain Kessi, Christian Kern, Jörg Knappen, Kjetil Kjernsmo, Maik Lehradt, Alexander Mai, Martin Maechler, Claus Malten, Kevin Van Maren, Lenimar Nunes de AndradeHubert Partl, John Reffing, Mike Ressler, Brian Ripley, Young U. Ryu, Bernd Rosenlecher, Chris Rowley, Hanspeter Schmid, Craig Schlenker, Christopher Sawtell, Josef Tkadlec, Didier Verna, Fabian Wernli, Carl-Gustav Werner, Chris York, Fritz Zaucker, Rick Zaccane, and Mikhail Zotov, #TeX.

Úvod

L^AT_EX [1] je typografický systém, ktorý je veľmi vhodný na výrobu vedeckých a matematických dokumentov vysokej typografickej kvality. Je tiež vhodný na prípravu všetkých druhov iných dokumentov, od jednoduchých listov až po celé knihy. L^AT_EX používa T_EX [2] ako formátovací jazyk.

Tento krátky úvod opisuje L^AT_EX 2_ε a mal by byť dostatočným pre väčšinu aplikácií L^AT_EXu. Podrobnejšie sa o L^AT_EXu môžete dočítať v [1, 3].

L^AT_EX je dostupný pre väčšinu počítačov od PC a Mac-ov až po veľké systémy UNIX a VMS. Na mnohých univerzitných sieťach je už inštalovaný a pripravený na prácu. Informácie o prístupe k lokálnej inštalácii L^AT_EX u by mali byť uvedené v *Local Guide* [4]. Ak máte problémy so spustením, obráťte sa na osobu, ktorá vám poskytla túto brožúru. Cieľom tohoto dokumentu *nie je* povedať vám ako nainštalovať a nastaviť systém L^AT_EX, ale naučiť vás, ako písať dokumenty tak, aby mohli byť spracované L^AT_EXom.

Tento úvod je rozdelený do piatich kapitol:

Kapitola 1 sa venuje základnej štruktúre dokumentov, písaných v L^AT_EXu.

Dozviete sa tiež niečo z histórie L^AT_EXu. Po prečítaní tejto kapitoly by ste mali mať približný obraz o L^AT_EXu. Bude to zatiaľ len rám obrazu, ale umožní vám to poskladať informácie z ďalších kapitol do veľkého obrazu.

Kapitola 2 rozvádza do podrobností písanie vášho dokumentu. Vysvetľuje najdôležitejšie L^AT_EXové príkazy a prostredia. Po prečítaní tejto kapitoly budete schopní napísať váš prvý dokument.

Kapitola 3 vysvetľuje ako písať v L^AT_EXu vzorce. Množstvo príkladov vám znova pomôže pochopiť jednu zo silných stránok L^AT_EXu. Na konci tejto kapitoly nájdete tabuľky, obsahujúce zoznamy všetkých matematických symbolov prístupných v L^AT_EXu.

Kapitola 4 vysvetľuje vytváranie indexu a zoznamu literatúry, vkladanie eps grafiky¹ do vášho dokumentu a niektoré ďalšie užitočné rozšírenia.

¹Obrázkov vo formáte ENCAPSULATED POSTSCRIPT (pozn. prekl.)

Kapitola 5 obsahuje určitú potenciálne nebezpečnú informáciu o tom ako dosiahnuť alternatívnu úpravu voči štandardnej úprave, ktorú vytvára L^AT_EX. Dozviete sa z nej ako zmeniť veci tak, že krásny výstup z L^AT_EXu začne vyzeráť dosť zle.

Je dôležité, aby ste čítali kapitoly po poradí. Okrem toho táto kniha nie je až taká veľká. Pozorne si prečítajte príklady, pretože veľká časť informácií je obsiahnutá práve v týchto príkladoch, ktoré nájdete v celej knihe.

Ak potrebujete akýkoľvek materiál súvisiaci s L^AT_EXom, pozrite sa do niektorého z CTAN ftp archívov. Pre USA je to `ftp.shsu.edu`, pre Nemecko `ftp.dante.de`, pre Veľkú Britániu `ftp.tex.ac.uk` alebo `ftp.muni.cz` u nás. Ak nie ste v niektorej z týchto krajín, zvolte si archív, ktorý je ku vám najbližšie.

Ak chcete spustiť L^AT_EX na vašom vlastnom počítači, pozrite sa čo sa dá získať z `CTAN:/tex-archive/systems`.

Ak vás napadne niečo, čo by mohlo byť pridané, vypustené, alebo zmenené v tomto dokumente, prosím, oznámte mi to. Názory L^AT_EXových začiatočníkov ma zaujímajú najviac – čo z tohoto úvodu (do L^AT_EXu) ľahko pochopili a čo by podľa nich mohlo byť vysvetlené lepšie.

Tobias Oetiker <oetiker@ee.ethz.ch>
Department of Electrical Engineering,
Swiss Federal Institute of Technology

Aktuálna verzia originálu tohoto dokumentu je prístupná na
`CTAN:/tex-archive/info/lshort`

Ján Buša st. <busaj@tuke.sk>
Katedra matematiky FEI,
Technická univerzita v Košiciach

Aktuálna verzia tohoto dokumentu je prístupná na
`www.cstug.sk`

Obsah

Vďaka!	iii
Úvod	v
1 Veci, ktoré potrebujete vedieť	1
1.1 The Name of the Game	1
1.1.1 T _E X	1
1.1.2 L ^A T _E X	1
1.2 Základy	3
1.2.1 Autor, knižný dizajnér a sadzač	3
1.2.2 Výzor výstupu	3
1.2.3 Výhody a nevýhody	4
1.3 Vstupné súbory L ^A T _E Xu	5
1.3.1 Medzery	5
1.3.2 Špeciálne znaky	5
1.3.3 Príkazy L ^A T _E Xu	6
1.3.4 Poznámky	7
1.4 Štruktúra zdrojového súboru	7
1.5 Úprava dokumentu	8
1.5.1 Štýly dokumentov	8
1.5.2 Balíky – packages	11
1.5.3 Štýly strán	11
1.6 Súbory, na ktoré môžete natrafiť	11
1.7 Veľké projekty	14
2 Sádzanie textu	17
2.1 Štruktúra textu a jazyka	17
2.2 Lámanie riadkov a strán	19
2.2.1 Zarovnávanie odstavcov	19
2.2.2 Delenie slov	20
2.3 Refazce pripravené na použitie	21
2.4 Špeciálne znaky a symboly	21
2.4.1 Úvodzovky	21

2.4.2	Pomlčky a rozdeľovníky	22
2.4.3	Vlnka (\sim)	22
2.4.4	Bodky (. . .)	22
2.4.5	Ligatúry	23
2.4.6	Diakritika a špeciálne znaky	23
2.5	Medzinárodná jazyková podpora	24
2.6	Medzery medzi slovami	25
2.7	Tituly, kapitoly a oddiely	25
2.8	Krížové odkazy	27
2.9	Poznámky pod čiarou	28
2.10	Zvýraznené slová	28
2.11	Prostredia	29
2.11.1	Odrážky, vyratovania a popisy	29
2.11.2	Vlajočková sadzba a centrovanie	29
2.11.3	Dlhšie citácie a verše	30
2.11.4	Doslovná tlač	31
2.11.5	Vytváranie tabuliek	32
2.11.6	Nastavenie tabulátorov	34
2.12	Plávajúce objekty	35
3	Sádzanie matematických vzorcov	39
3.1	Všeobecné	39
3.2	Zoskupovanie v matematickom režime	41
3.3	Sádzanie matematických výrazov	41
3.4	Medzery v matematike	45
3.5	Vertikálne zarovnaný materiál	46
3.6	Duch	48
3.7	Veľkosť matematického fondu	48
3.8	Vety, zákony,	49
3.9	Tučné znaky	51
3.10	Zoznam matematických symbolov	52
4	Špeciality	59
4.1	Vkladanie EPS grafiky	59
4.2	Prostredie <code>picture</code>	61
4.3	Zoznam použitej literatúry	63
4.4	Tvorba registrov	64
4.5	Exotické hlavičky	65
4.6	Balík verbatim	65
4.7	Ochrana krehkých príkazov	66

5 Prispôsobovanie L^AT_EXu	69
5.1 Nové príkazy, prostredia a balíky	69
5.1.1 Nové príkazy	70
5.1.2 Nové prostredia	71
5.1.3 Váš vlastný balík	71
5.2 Fonty a veľkosti	72
5.2.1 Príkazy na zmenu písma	72
5.2.2 Pozor, nebezpečenstvo!	75
5.2.3 Rada	75
5.3 Medzery	76
5.3.1 Riadkovanie	76
5.3.2 Formátovanie odstavcov	76
5.3.3 Vodorovné medzery	77
5.3.4 Zvislé medzery	77
5.4 Rozvrhnutie stránky	78
5.5 Viac zábavy s dĺžkami	80
5.6 Škatuľky, krabičky, schránky, skrinky, boxy	81
5.7 Čiary a rozpery	83
Literatúra	85
Register	87

Zoznam obrázkov

1.1	Zložky systému T _E X.	2
1.2	Minimálny L ^A T _E Xový súbor	8
1.3	Príklad realistického časopiseckého článku	9
4.1	Príklad použitia prostredia <code>picture</code>	62
4.2	Príklad nastavenia <code>fancyhdr</code>	66
5.1	Príklad balíka.	72
5.2	Parametre ovplyvňujúce rozvrhnutie stránky	79

Zoznam tabuliek

1.1	Štýly (triedy) dokumentov	9
1.2	Nastavenia dokumentu (voľby príkazu <code>\documentclass</code>) . . .	10
1.3	Niektoré balíky, distribuované s \LaTeX om	12
1.4	Preddefinované štýly strán \LaTeX u	13
2.1	Diakritika a špeciálne znaky	23
2.2	Podmienky na umiestnenie plávajúceho objektu	36
3.1	Diakritika v matematickom režime	52
3.2	Malé grécke písmená	52
3.3	Veľké grécke písmená	52
3.4	Binárne relácie	53
3.5	Binárne operátory	53
3.6	Veľké operátory	54
3.7	Šípky	54
3.8	Oddeľovače	54
3.9	Veľké oddeľovače	54
3.10	Rôzne znaky	55
3.11	Nematematické znaky	55
3.12	AMS oddeľovače	55
3.13	AMS grécke a hebrejské znaky	55
3.14	AMS binárne relácie	56
3.15	AMS šípky	56
3.16	AMS negované binárne relácie a šípky	57
3.17	AMS binárne operátory	57
3.18	AMS rôzne	58
3.19	Matematické abecedy	58
4.1	Názvy kľúčov pre balík <code>graphicx</code>	60
4.2	Príklady syntaxe indexového kľúča	64
5.1	Fonty	73
5.2	Veľkosti fontov	74
5.3	Matematické fonty	74
5.4	Jednotky \TeX u	78

Kapitola 1

Veci, ktoré potrebujete vedieť

V prvej časti tejto kapitoly sa dozviete trochu o filozofii a histórii \LaTeX u 2_ε. Druhá časť kapitoly sa venuje základnej štruktúre \LaTeX ového dokumentu. Po prečítaní tejto kapitoly by ste mali mať hrubú predstavu o tom, ako pracuje \LaTeX . Prečítanie tejto kapitoly vám pomôže poskladať všetky ďalšie informácie do veľkého obrazu.

1.1 The Name of the Game

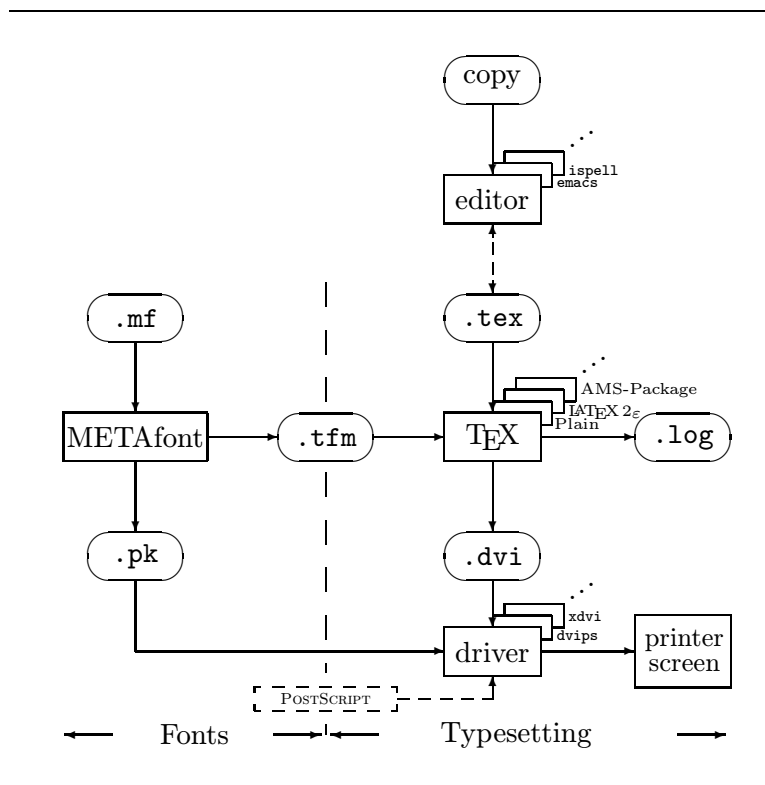
1.1.1 \TeX

\TeX je počítačový program, ktorého autorom je Donald E. Knuth [2]. Je určený na kvalitnú tlač textu a matematických vzorcov. Knuth začal písať typografický nástroj \TeX v roku 1977 aby preskúmal potenciál digitálneho tlačového vybavenia, ktoré v tom čase začalo pomaly prenikať do typografického priemyslu, najmä v nádeji, že dokáže zvrátiť trend zhoršujúcej sa typografickej kvality postihujúci jeho vlastné knihy a články. \TeX , ktorý používame dnes, bol uvoľnený v roku 1982, s niekoľkými drobnými rozšíreniami pridanými v roku 1989 na lepšiu podporu 8-bitových písmen a využitie mnohonásobných jazykov. \TeX je slávny svojou extrémnou stabilitou, schopnosťou použitia na mnohých rôznych typoch počítačov a skutočnou bezchybnosťou. Číslo verzie konverguje k π a v súčasnosti je to 3.14159.

\TeX sa číta ako „Tech“, s „ch“ ako v nemeckom slove „Ach“ alebo škótskom „Loch“. V prostredí ASCII sa \TeX píše TeX .

1.1.2 \LaTeX

\LaTeX je balík makier, ktorý umožňuje autorovi písať a následne tlačiť svoje práce v najvyššej typografickej kvalite, používajúc preddefinované profe-

Obr. 1.1: Zložky systému \TeX .

sionálne rozmiestnenie obsahu dokumentu. Pôvodným autorom \LaTeX u je Leslie Lamport [1]. Používa \TeX ako typografický nástroj.

V roku 1994 bol balík \LaTeX aktualizovaný $\text{\LaTeX}3$ teamom, vedeným Frankom Mittelbachom tak, aby obsahoval niektoré dlho požadované vylepšenia a aby sa zjednotili všetky záplatové verzie, ktoré vznikli od vytvorenia \LaTeX u 2.09 pred niekoľkými rokmi. Nato, aby sa nová verzia dala odlíšiť od starej, bola pomenovaná $\text{\LaTeX}2_{\epsilon}$. Táto publikácia sa zaoberá práve už spomínanou verziou $\text{\LaTeX}2_{\epsilon}$.

\LaTeX sa vyslovuje „Lay-tech“ alebo „Lah-tech“. \LaTeX sa v prostredí ASCII píše \LaTeX . $\text{\LaTeX}2_{\epsilon}$ sa vyslovuje „Lej-tech dva e“ a píše sa $\text{\LaTeX}2_{\epsilon}$.

Hore uvedený obrázok 1.1 ukazuje spoluprácu \TeX u a $\text{\LaTeX}2_{\epsilon}$ u. Tento obrázok je prevzatý z `wots.tex` od Kees van der Laana.

1.2 Základy

1.2.1 Autor, knižný dizajnér a sadzač

Na opublikovanie niečoho odnesú autori svoj rukopis do vydavateľstva. Knižný dizajnér vydavateľstva potom rozhodne o úprave dokumentu (o šírke stĺpcov, druhoch písma, medzerách pred a po nadpisoch, ...). Dizajnér napíše inštrukcie do rukopisu a dá ho sadzačovi, ktorý pripraví knihu podľa týchto inštrukcií.

Ľudský dizajnér sa snaží zistiť, čo si autor myslel počas písania rukopisu. Na základe svojich profesionálnych vedomostí a obsahu rukopisu rozhoduje o nadpisoch kapitol, citáciách, príkladoch, vzorcoch, atď.

V prostredí \LaTeX u berie na seba úlohu dizajnéra \LaTeX a používa \TeX ako svojho sadzača. Ale \LaTeX je tiež „len“ program a preto potrebuje ďalšie vedenie. Autor musí poskytnúť dodatočné informácie, ktoré opisujú štruktúru jeho práce. Tieto informácie sa vpisujú do textu ako „ \LaTeX ové príkazy“.

Toto je trochu odlišné od prístupu WYSIWYG¹, ktorý využíva väčšina moderných procesorov, takých ako *MS Word* alebo *Corel WordPerfect*. Pri týchto aplikáciách, autori určujú výstup interaktívne počas písania textu do počítača. Počas celej práce takto môžu vidieť na monitore, ako bude práca nakoniec vyzeráť, keď bude vytlačena.

Počas používania \LaTeX u normálne nie je možné počas písania vidieť konečný výstup. Konečný tvar textu sa však môže prezerať na monitore po spracovaní textu \LaTeX om. Potom sa ďalšie úpravy môžu urobiť pred vlastnou tlačou dokumentu.

1.2.2 Výzor výstupu

Typografický dizajn je remeslo. Neskúsení autori sa často dopúšťajú vážnych chýb, mysliac si, že výzor knihy je hlavne vec estetiky—„Ak dokument vyzerá dobre po umeleckej stránke, potom je navrhnutý“. Keďže však má dokument slúžiť na čítanie a nie visieť niekde v galérii, čitateľnosť a zrozumiteľnosť sú najdôležitejšie a až potom nasleduje pekný výzor. Príklady:

- veľkosť textu a číslovanie kapitol sa má vyberať tak, aby štruktúra kapitol a odsekov bola jasná pre čitateľa,
- riadky majú byť dostatočne krátke, aby nenamáhal zrak čitateľa a zároveň dosť dlhé, aby pekne zaplnili stranu.

Použitím systémov WYSIWYG autori často dosahujú esteticky zrak potešujúce dokumenty, s nedostatočnou alebo nedôslednou štruktúrou. \LaTeX predchádza takýmto chybám pri formátovaní, nútiac autora vysvetliť *logickú*

¹What you see is what you get. – Čo vidíš, to dostaneš.

štruktúru svojho dokumentu. \LaTeX následne zvolí výstup, najviac zodpovedajúci požiadavkam autora.

1.2.3 Výhody a nevýhody

Keď ľudia z WYSIWYGového sveta stretnú ľudí, ktorí používajú \LaTeX , najčastejšie diskutujú „výhody \LaTeX u v porovnaní s normálnym textovým procesorom“ alebo naopak. Najlepšie urobíte, ak takýto rozhovor začne, keď budete v úzadí, keďže sa takáto diskusia často vymyká z rúk. Avšak niekedy niet úniku . . .

Teda tu je troška streliva. Najväčšie výhody \LaTeX u oproti normálnym textovým procesorom sú nasledujúce:

- Sú dostupné profesionálne vytvorené úpravy stránky, takže dokument skutočne vyzerá, ako keby bol „vytlačенý“.
- Písanie matematických vzorcov je zabezpečené pohodlným spôsobom.
- Užívateľovi sa stačí naučiť niekoľko ľahko pochopiteľných príkazov, definujúcich logickú štruktúru dokumentu. Takmer nikdy nemusí fušovať do nastavenej úpravy dokumentu.
- Ľahko môžu byť vygenerované aj zložité štruktúry ako poznámky pod čiarou, odkazy, obsah, použitá literatúra a index.
- Na riešenie mnohých typografických úloh, základným \LaTeX om priamo nepodporovaných, existujú voľne pripojiteľné balíky makier. Napríklad môžete pomocou nich vložiť do textu POSTSCRIPTOVÚ grafiku alebo vytvoriť bibliografie, dodržiujúc presné štandardy. Mnoho z týchto voľne pripojiteľných makier je opísaných v *The \LaTeX Companion* [3].
- \LaTeX povzbudzuje autorov písať texty s kvalitnou štruktúrou, pretože toto je spôsob, ako \LaTeX pracuje — špecifikovaním štruktúry.
- \TeX , formátovací motor $\LaTeX 2_{\epsilon}$, je bezplatný a ľahko prenositeľný. Preto beží takmer na každej dostupnej hardverovej platforme.

\LaTeX má aj určité nevýhody, ale myslím si, že je pre mňa trochu obtiažne nájsť nejakú rozumnú, hoci som si istý, že iní ľudia by vám mohli povedať o stovkách ; -)

- \LaTeX nefunguje dobre ľuďom, ktorí predali svoju dušu . . .
- Hoci niektoré parametre preddefinovaných foriem strán sa môžu upravovať, dizajn celého nového formátu je ťažký a zaberie veľa času².

²Rumor hovorí, že to bude jeden z kľúčových prvkov, adresovaných pripravovanému systému $\LaTeX 3$

- Je veľmi ťažké napísať neštrukturovaný a chaotický dokument.
- Váš škrečok by nemusel byť nikdy schopný, navzdory niektorým povzbudzujúcim prvým krokom, úplne pochopiť pojem logického značkovania (Logical Markup)

1.3 Vstupné súbory \LaTeX u

Zdrojový súbor pre \LaTeX je jednoduchý súbor v ASCII. Môžete ho vytvoriť ľubovoľným textovým editorom. Obsahuje text dokumentu rovnako ako aj príkazy, ktoré hovoria \LaTeX u ako vysádzať text.

1.3.1 Medzery

Znaky „bielych miest“, ako medzera alebo tabulátor, sa v \LaTeX u interpretujú zhodne ako medzera („space“). *Viacero za sebou nasledujúcich* medzier sa považuje za *jednu* medzeru. Voľné miesto na začiatku riadku sa vo všeobecnosti ignoruje a samotný koniec riadku sa považuje za jednu medzeru.

Prázdny riadok medzi dvoma riadkami textu definuje koniec odstavca. *Viacero* voľných riadkov sa berie rovnako ako *jeden* voľný riadok. Príkladom je nižšie uvedený text. Na ľavej strane vidíte zdrojový text a na pravej strane je preformátovaný výstup.

Nezáleží na tom, či
zadáte za slovom jednu,
alebo viac medzier.

Voľný riadok začína nový
odstavec.

Nezáleží na tom, či zadáte za slovom jednu,
alebo viac medzier.

Voľný riadok začína nový odstavec.

1.3.2 Špeciálne znaky

Nasledujúce znaky sú rezervovanými symbolmi, ktoré buď majú pre \LaTeX zvláštny význam alebo nie sú dostupné vo všetkých fontoch. Ak ich napíšete priamo do textu, zvyčajne sa pri tlači nevytlačia, ale väčšinou donútia \LaTeX robiť také veci, ktoré ste nemali v úmysle.

$\$ \& \% \# _ \{ \} \sim \hat \ \backslash$

Ako uvidíte, tieto symboly môžete vložiť do vášho dokumentu pridaním obráteného lomítka pred ne:

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

`$ & % # - { }`

Zvyšné symboly (a mnoho ďalších) môžu byť vytlačené pomocou špeciálnych príkazov v matematickom prostredí alebo ako diakritické znamienka. Znak obráteného lomítka `\` *nemôžeme* získať pridaním ďalšieho spätného lomítka pred neho (`\\`), táto postupnosť sa používa na zalamovanie riadkov.³

1.3.3 Príkazy \LaTeX u

Príkazy \LaTeX u rozlišujú veľké a malé písmená a majú jeden z nasledujúcich dvoch tvarov:

- Začínajú sa obráteným lomítkom `\` a potom pokračujú názvom zloženým len z písmen. Názvy príkazov sú zakončené medzerou, číslom alebo nejakým iným „nepísmenom“.
- Skladajú sa z obráteného lomítka a práve jedného znaku (číselného alebo špeciálneho).

\LaTeX ignoruje medzery za príkazmi. Ak chcete dostať za príkazom medzeru, musíte za príkaz napísať `{ }` a medzeru, alebo použiť špeciálny príkaz po zadaní vášho príkazu. Zátvorky (`{ }`) zabránia \LaTeX u požíerať všetky medzery za názvom príkazu.

Čítal som, že Knuth rozdeľuje
ľudí používajúcich `\TeX{}` na
`\TeX{}`nikov a `\TeX` pertov. `\\`
Dnes je `\today`.

Čítal som, že Knuth rozdeľuje ľudí používa-
júcích \TeX na \TeX nikov a \TeX pertov.
Dnes je 2. mája 2001.

Niektoré príkazy vyžadujú parameter, ktorý sa vkladá do zložených zátvoriek `{ }` za názvom príkazu. Niektoré príkazy podporujú nepovinné parametre, ktoré sa pridávajú za názov príkazu do hranatých zátvoriek `[]`. Nasledujúci príklad používa niektoré \LaTeX ové príkazy. Netrápte sa kvôli nim, neskôr budú vysvetlené.

Môžete ma `\textsl{naklonit}`!

Môžete ma *naklonit*!

³Vyskúšajte namiesto toho príkaz `\backslash$`. Vytvorí znak `'\'`.

Začnite, prosím, nový riadok
 práve tu!\newline
 Ďakujem!

Začnite nový riadok
 práve tu!\linebreak[3]
 Ďakujem!

Začnite nový riadok
 práve tu!\linebreak[4]
 Ďakujem!

Začnite, prosím, nový riadok práve tu!
 Ďakujem!
 Začnite nový riadok práve tu! Ďakujem!
 Začnite nový riadok práve tu!
 Ďakujem!

1.3.4 Poznámky

Ak \LaTeX narazí počas spracovávania vstupného súboru na znak percenta %, bude ignorovať zvyšok daného riadku, znak ukončenia riadku a všetky medzery na začiatku nasledujúceho riadku.

Toto sa dá použiť na písanie poznámok do vstupného súboru, ktoré sa neukážu vo vytlačenej verzii.

Toto je % hlúpy
 % Lepšie: poučný <----
 príklad: super%
 predí%
 žené slovo

Toto je príklad: superpredĺžené slovo

Znak % môže byť tiež použitý na rozdelenie dlhých vstupných riadkov, v ktorých sú zakázané medzery, resp. konce riadkov.

Na dlhšie poznámky musíte použiť prostredie `comment` poskytované balíkom `verbatim`.

Toto je ďalší
`\begin{comment}`
 dost hlúpy,
 ale užitočný
`\end{comment}`
 príklad začlenenia komentárov do vášho dokumentu.

Toto je ďalší príklad začlenenia komentárov do vášho dokumentu.

1.4 Štruktúra zdrojového súboru

Keď \LaTeX 2_{ϵ} spracúvava zdrojový súbor, očakáva, že bude mať určitú štruktúru. Teda každý zdrojový súbor musí začínať príkazom

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

Toto špecifikuje, aký štýl dokumentu chcete napísať. Potom môžete zadať príkazy, ktoré majú ovplyvniť štýl celého dokumentu, alebo môžete natiahnuť balíky (makrá), ktoré pridajú do systému L^AT_EX nové vybavenie. Na natiahnutie takýchto balíkov použijete príkaz

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

Keď je všetko nastavené⁴, začnete samotný text príkazom

```
\begin{document}
```

Ďalej napíšete text, pomešovaný s niekoľkými užitočnými L^AT_EXovými príkazmi. Na konci dokumentu pridáte príkaz

```
\end{document}
```

ktorý povie L^AT_EXu, že tu končí váš text. Všetko, čo nasleduje za týmto príkazom, bude L^AT_EX ignorovať.

Obrázok 1.2 ukazuje obsah minimálneho L^AT_EXového súboru, trochu zložitejší zdrojový súbor ukazuje obrázok 1.3.

```
\documentclass{article}
\usepackage{slovak}
\begin{document}
Malé je pekné.
\end{document}
```

Obr. 1.2: Minimálny L^AT_EXový súbor

1.5 Úprava dokumentu

1.5.1 Štýly dokumentov

Prvá informácia, ktorú potrebuje L^AT_EX počas formátovania súboru, je typ dokumentu, aký chcel autor vytvoriť. To definuje príkaz `\documentclass`:

```
\documentclass[options]{class}
```

Parameter *class* špecifikuje štýl (triedu) dokumentu, ktorý chceme vytvoriť. Tabuľka 1.1 uvádza štýly dokumentov, ktoré sú v tejto príručke obsiahnuté. V distribúcii L^AT_EX 2_ε sa stretávame aj s ďalšími triedami pre iné dokumenty,

⁴Priestor medzi `\documentclass` a `\begin{document}` sa nazýva *preambula*. Ak používate v C^ST_EXu diakritiku, musíte zadať príkaz `\usepackage{slovak}`, resp. `\usepackage{czech}`.

vrátane listov a fólií. Parameter *options* upravuje základné nastavenia dokumentu. Voľby musia byť oddelené čiarkami. Najbežnejšie voľby (nastavenia) pre štandardné štýly dokumentu nájdete v tabuľke 1.2.

Príklad: Vstupný súbor pre L^AT_EX môže začínať riadkom

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

Tento riadok prikazuje L^AT_EXu vysádzať dokument ako *článok* so základnou veľkosťou fontov *jedenásť bodov*, pričom výstup musí byť vhodný na

```
\documentclass[a4paper,11pt]{article}
\usepackage{slovak}
\usepackage{latexsym}
\author{H.~Partl}
\title{Minimalizmus}
\frenchspacing
\begin{document}
\maketitle
\tableofcontents
\section{Začiatok}
Nuž a tu začína môj prekrásny článok.
\section{Koniec}
\ldots{} a tu končí.
\end{document}
```

Obr. 1.3: Príklad realistického časopiseckého článku

Tabuľka 1.1: Štýly (triedy) dokumentov

article na články vo vedeckých časopisoch, prezentácie, krátke správy, dokumentáciu programu, pozvánky, ...

report na dlhšie správy, obsahujúce niekoľko kapitol, krátke knihy, dizertačné práce, ...

book na skutočné knihy

letter na písanie listov (korešpondenciu)

slides na fólie. Tento typ používa veľké bezpätkové písmená (*sans serif*). Namiesto toho sa môžete rozhodnúť pre používanie FoilT_EX^a.

^aCTAN:/tex-archive/macros/latex/packages/supported/foiltex

Tabuľka 1.2: Nastavenia dokumentu (voľby príkazu `\documentclass`)

-
- `10pt`, `11pt`, `12pt` — nastavuje veľkosť hlavného fondu, použitého v dokumente. Ak sa voľba nepoužije, predpokladá sa `10pt`.
- `a4paper`, `letterpaper`, ... — definuje formát strany. Východzia veľkosť je `letterpaper`. Okrem toho sa môžu použiť aj formáty `a5paper`, `b5paper`, `executivepaper` a `legalpaper`.
- `fleqn` — namiesto centrovania zobrazí vzorce (rovnice) zarovnané doľava.
- `leqno` — umiestňuje číslovanie vzorcov (rovníc) na ľavú stranu (vzorca) namiesto pravej.
- `titlepage`, `notitlepage` — špecifikuje, či po názve dokumentu má byť začatá nová strana, alebo nie. Pri dokumente typu `article` nie je implicitne nastavené, aby po názve bola začatá nová strana, kým pri triedach `report` a `book` to tak nastavené je.
- `twocolumn` — hovorí $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ u, aby vysádzal dokument v dvoch stĺpčekoch.
- `twoside`, `oneside` — upresňuje, či výstup má byť jedno- alebo obojstranný. Pri triedach `article` a `report` je základom jednostranná a pri štýle `book` je to obojstranná tlač. Voľba `twoside` *neoznami* vašej tlačiarni, že má skutočne vyrobiť dvojstranný výstup.
- `openright`, `openany` — zabezpečí, aby kapitoly začínali buď iba na pravej strane, alebo na ktorejkoľvek nasledujúcej voľnej strane. Toto nastavenie nepracuje pri štýle `article`, pretože tento typ dokumentu nepozná kapitoly. Pri dokumentoch typu `report` štandardne začína nová kapitola na najbližšej voľnej stránke a pri triede `book` na najsledujúcej pravej strane.
-

obojsmernú tlač na papieri formátu *A4*.

1.5.2 Balíky – packages

Počas písania dokumentu pravdepodobne nájdete miesta, kde samotný základný \LaTeX nemôže vyriešiť váš problém. Keď chcete do vášho dokumentu vložiť grafiku, farebný text alebo zdrojový text zo súboru, musíte zväčšiť schopnosti \LaTeX u. Takéto rozšírenia sa nazývajú balíky makier. Tieto balíky sa aktivujú pomocou príkazu

```
\usepackage[options]{package}
```

kde *package* je názov balíka a *options* je zoznam kľúčových slov, ktoré spúšťajú špeciálne vlastnosti daného balíka. Niektoré balíky sú obsiahnuté v základnej distribúcii \LaTeX 2 ϵ (pozri tabuľku 1.3). Ďalšie sú šírené samostatne. Viac informácií o balíkoch, inštalovaných na vašej užívateľskej stanici, sa môžete dočítať vo vašom *Local Guide* [4]. Základným zdrojom informácií o \LaTeX u je *The \LaTeX Companion* [3]. Obsahuje popisy stoviek balíkov, spolu s informáciami, ako napísať vaše vlastné rozšírenia pre \LaTeX 2 ϵ .

1.5.3 Štýly strán

\LaTeX podporuje tri preddefinované kombinácie záhlavia/päty strán, nazývané štýly strán (page styles). Príkaz

```
\pagestyle{style}
```

definuje, ktorú z možných kombinácií treba použiť. Tabuľka 1.4 uvádza preddefinované štýly.

Štýl aktuálnej stránky sa môže meniť príkazom

```
\thispagestyle{style}
```

V *The \LaTeX Companion* [3] a v oddieli 4.5 na strane 65 nájdete popis, ako si môžete vytvoriť vaše vlastné záhlavie a päť strany.

1.6 Súbory, na ktoré môžete natrafiť

Pri práci s \LaTeX om sa čoskoro ocitnete v bludisku súborov s rôznymi koncovkami (extensions) a pravdepodobne bez pochopenia ich pôvodu. Dole uvedený zoznam vám napovie niečo o rôznych typoch súborov, na ktoré môžete natrafiť pri práci s \TeX om. Prosím, vezmite však na vedomie, že táto tabuľka si nenárokujú stať sa úplným zoznamom koncoviek a ak nájdete takú, o ktorej ste presvedčení, že je dôležitá, napíšte nám pár riadkov.

 Tabuľka 1.3: Niektoré balíky, distribuované s L^AT_EXom

`doc` umožňuje dokumentáciu L^AT_EXových programov.

Popis je v `doc.dtx`^a a v *The L^AT_EX Companion* [3].

`exscale` poskytuje stupňované verzie rozšíreného matematického fontu.

Popis v `ltxscale.dtx`.

`fontenc` stanovuje, aké kódovanie fontov má L^AT_EX použiť.

Popis v `ltoutenc.dtx`.

`ifthen` poskytuje príkazy typu ‘ak ... potom urob ... ináč urob ...’.

Popis v `ifthen.dtx` a v *The L^AT_EX Companion* [3].

`latexsym` musíte použiť na sprístupnenie fontu symbolov L^AT_EXu.

Popis v `latexsym.dtx` a v *The L^AT_EX Companion* [3]

`makeidx` poskytuje príkazy na tvorbu indexov.

Popis v kapitole 4.4 a v *The L^AT_EX Companion* [3].

`syntonly` spracuje dokument bez toho, aby ho vysádzal.

Popis v `syntonly.dtx` a v *The L^AT_EX Companion* [3]. Toto je užitočné na rýchle preverenie chýb.

`inputenc` umožňuje špecifikovať vstupné kódovanie ako ASCII, ISO Latin-1, ISO Latin-2, kódovanie strán 437/850 IBM, Apple Macintosh, Next, ANSI-Windows alebo definované užívateľom.

Popis v `inputenc.dtx`.

^aTento súbor by mal byť inštalovaný na vašom systéme a mali by ste mať možnosť získať `doc.dtx` pomocou príkazu `latex doc.dtx`. Toto isté platí aj o ostatných balíkoch, uvádzaných v tejto tabuľke

- .tex** Vstupný (zdrojový) súbor \LaTeX u alebo \TeX u. Môže ho kompilovať `latex`, resp. `tex`.
- .sty** \LaTeX ovský balík makier. Je to súbor, ktorý môžete zaradiť do vášho \LaTeX ovského dokumentu príkazom `\usepackage`.
- .dtx** Dokumentovaný \TeX . Je to základný formát rozširovaných \LaTeX ových štýlových súborov. Ak spracujete `.dtx` súbor, získate dokumentovaný kód makier \LaTeX ovského balíka, obsiahnutého v `.dtx` súbore.
- .ins** Je to inštalátor pre súbory obsiahnuté v označenom `.dtx` súbore. Ak si stiahnete \LaTeX ovský balík zo siete, väčšinou získate súbory `.dtx` a `.ins`. Spustením \LaTeX u so súborom `.ins` rozbalíte súbor `.dtx`.
- .cls** Súbory tried určujú vzhľad vašich dokumentov. Zavádzajú sa príkazom `\documentclass`.

Nasledujúce súbory sa vygenerujú ak spustíte \LaTeX so svojim zdrojovým súborom:

- .dvi** Device Independent file (súbor nezávislý na zariadení). Toto je hlavný výsledok kompilovania \LaTeX om. Jeho obsah si môžete pozrieť pomocou DVI-prehliadača (DVI previewer program) alebo ho môžete poslať do tlačiarne po použití programu `dvips` alebo podobnej aplikácie.
- .log** Podáva podrobnú správu o priebehu poslednej kompilácie.
- .toc** Sem sa ukladajú hlavičky oddielov. Načíta sa pri nasledujúcej kompilácii a použije sa na vytvorenie obsahu (table of content).
- .lof** Podobne ako `.toc` súbor, avšak obsahuje zoznam obrázkov (list of figures).
- .lot** A znova to isté pre zoznam tabuliek (list of tables).

Tabuľka 1.4: Preddefinované štýly strán \LaTeX u

plain vytlačí čísla strán na spodok strany do stredu päty strany. Toto je štandardný štýl strany.

headings vytlačí do záhlavia každej strany názov aktuálnej kapitoly (oddielu) a číslo strany, pričom päta dokumentu ostane prázdna. (Takýmto štýlom je napísaný tento dokument.)

empty nastavuje, aby záhlavie ako aj päta stránky ostali prázdne.

- .aux** Ďalší súbor prenášajúci informáciu z jednej kompilácie do ďalšej. Okrem iného sa súbor `.aux` používa na uloženie informácií súvisiacich s krížnymi odkazmi.
- .idx** Ak váš dokument obsahuje register. \LaTeX sem ukladá všetky slová, ktoré sa majú objaviť v registri. Tento súbor musí byť spracovaný programom `makeindex`. Kvôli ďalšej informácii o tvorbe indexu pozri oddiel 4.4 na strane 64.
- .ind** Je to výsledok spracovania súboru `.idx`, pripravený na zaradenie do vášho dokumentu počas nasledujúceho kompilačného cyklu.
- .ilg** Súbor hlásení o priebehu programu `makeindex`.

1.7 Veľké projekty

Počas práce s veľkými dokumentmi môžete chcieť rozdeliť vstupný súbor na niekoľko častí. \LaTeX má dva príkazy, ktoré vám s tým pomôžu.

Príkaz

```
\include{filename}
```

môžete použiť v texte na vloženie obsahu iného súboru *filename.tex*. Zapamätajte si, že \LaTeX pred spracovaním vloženého materiálu z *filename.tex* začne novú stranu.

Druhý príkaz môže byť použitý v preambule. Umožňuje vám to prikázať \LaTeX u, aby vložil len niektoré „include“ súbory.

```
\includeonly{filename, filename, ... }
```

Po umiestnení tohoto príkazu v preambule dokumentu, budú vykonané iba tie príkazy `\include`, ktoré sú uvádzané v `\includeonly`. Zapamätajte si, že v názve súboru (*filename*) sa nesmie vyskytovať čiarka ani medzera.

Príkaz `\include` začína tlač vkladaneho textu na novej strane. Toto je nápomocné, keď používate `\includeonly`, pretože zalamovania strán (*pagebreaks*) sa neposúvajú ani v prípade, ak niektoré z vkladaneých súborov vynecháme. Niekedy to však nemusí byť žiadúce. V takom prípade môžete použiť príkaz

```
\input{filename}
```

Jednoducho vloží do dokumentu špecifikovaný súbor. Nepridávajú sa žiadne reťazce.

Ak chcete, aby \LaTeX rýchlo skontroloval váš dokument, môžete použiť balík `syntonly`. Tento donúti \LaTeX pri prechádzaní vašim dokumentom kontrolovať len správnosť syntaxe a použitia príkazov, nevytvára sa však žiaden (DVI) výstup. Keďže pri tomto režime beží \LaTeX rýchlejšie, môžete ušetriť váš drahocenný čas. Použitie je veľmi jednoduché:

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

Ak chcete vytvoriť strany, stačí zakomentovať druhý riadok (pridaním znaku percenta).

Kapitola 2

Sádzanie textu

Po prečítaní predchádzajúcej kapitoly by ste mali vedieť o základnom materiále, z ktorého je dokument v \LaTeX vyrobený. V tejto kapitole doplním zvyšnú štruktúru, ktorú potrebujete poznať na tvorbu materiálov zo skutočného sveta.

2.1 Štruktúra textu a jazyka

Hlavným cieľom napísania textu (ak vylúčime určitú súčasnú literatúru typu DACC¹), je vyjadrenie myšlienok, sprostredkovanie informácie, alebo vedomostí čitateľovi. Čitateľ text lepšie pochopí, ak budú myšlienky dobre štruktúrované a uvidí a precíti túto štruktúru omnoho lepšie, ak bude typografická forma odrážať logickú a sémantickú štruktúru obsahu.

\LaTeX sa od iných typografických systémov líši práve tým, že mu musíte oznámiť logickú a sémantickú štruktúru textu. Potom odvodí typografickú formu na základe „pravidiel“ zadaných v súbore, definujúcom triedu a v rozličných štýlových súboroch.

Najdôležitejšia jednotka textu v \LaTeX (a v typografii) je odstavec (paragraph). Nazývame ho „textovou jednotkou“, pretože odstavec je typografická forma, ktorá by mala odrážať jednu súvislú myšlienku alebo jeden pojem. V nasledujúcich oddieloch sa naučíte ako si môžete vynútiť zalomenie riadku, napr. pomocou `\` a ukončenie odstavca napr. vynechaním prázdneho riadku v zdrojovom kóde. Preto, ak sa začína nová myšlienka, mal by sa začať aj nový odstavec a ak nie, mali by byť použité len zalomenia riadkov. Ak ste na pochybách ohľadom ukončení odstavcov, predstavte si svoj text ako sprostredkovateľa pojmov a myšlienok. Ak máte ukončený odstavec, ale stará myšlienka pokračuje, mal by byť nový odstavec zrušený. Ak sa v tom istom odstavci vyskytne nejaký úplne nový tok myšlienok, tak by odstavec mal byť rozdelený.

Väčšina ľudí úplne podceňuje dôležitosť správneho členenia odstavcov. Mnohí dokonca ani nepoznajú zmysel ukončenia odstavca alebo, najmä

¹Different At All Cost, preklad švajčiarsko-nemeckého UVA (Um's Verrecken Anders).

v \LaTeX u, ukončujú odstavce bez toho, aby si to uvedomovali. Táto druhá chyba sa dá obzvlášť jednoducho urobiť pri použití rovníc v texte. Pozrite si nasledujúce príklady a premyslite si, prečo sú prázdne riadky (ukončenia odstavcov) niekedy použité pred a za rovnicami a niekedy nie sú použité. (Ak ešte dosť dobre nechápete všetky príkazy, aby ste porozumeli týmto príkladom, prečítajte si, prosím, túto a nasledujúcu kapitolu a potom si prečítajte tento oddiel znova.)

```
% Example 1
\ldots keď Einstein zaviedol svoj vzťah
\begin{equation}
e = m \cdot c^2 \ ; \ ,
\end{equation}
ktorý je zároveň najznámejším a najmenej
pochopeným fyzikálnym vzorcom.
```

```
% Example 2
\ldots z ktorého vyplýva Kirchoffov zákon pre prúdy:
\begin{equation}
\sum_{k=1}^n I_k = 0 \ ; \ .
\end{equation}
```

Kirchoffov zákon pre napätia môže byť odvodený \ldots

```
% Example 3
\ldots ktorý má mnohé výhody.

\begin{equation}
I_D = I_F - I_R
\end{equation}
je jadro veľmi odlišného tranzistorového modelu. \ldots
```

Ďalšia menšia textová jednotka je veta. V anglických textoch sa dáva väčšia medzera za bodku, ktorá ukončuje vetu ako za bodku, ktorá sa používa na konci skratky. \LaTeX sa pokúša určiť, ktorú z nich chcete mať. Ak to \LaTeX urobí nesprávne, musíte mu povedať, čo chcete. Toto bude vysvetlené neskôr v tejto kapitole.

Štruktúrovanie textu pokračuje dokonca do častí viet. Väčšina jazykov ma veľmi zložité interpunkčné pravidlá, ale v mnohých jazykoch (vrátane nemčiny a angličtiny) dostanete takmer všetky čiarky správne ak si zapamätáte, čo reprezentujú: krátku zastávku v toku reči. Ak si nie ste istí v tom, kam umiestniť čiarku, prečítajte vetu nahlas a zľahka sa nadýchnite pri každej čiarku. Ak to na niektorom mieste pôsobí nemotorne, zrušte túto čiarku,

ak cítite potrebu nadýchnuť sa (alebo urobiť krátku prestávku) na nejakom inom mieste, vložte tam čiarku.

Nakoniec odstavce textu by mali byť logicky štruktúrované na vyššej úrovni tým, že sú zoskupované do kapitol, oddielov, pododdielov, atď. Nakoniec typografický účinok napísania napr. `\section{Štruktúra textu a jazyka}` je tak jasný, že je takmer úplne evidentné ako majú byť tieto štruktúry vyššej úrovne používané.

2.2 Lámanie riadkov a strán

2.2.1 Zarovnávanie odstavcov

Väčšinou sa knihy tlačia tak, že každý riadok má rovnakú dĺžku. \LaTeX láme riadky a vkladá medzery medzi slovami tak, aby dosiahol optimálny vzhľad celého odstavca. V prípade, že je to nutné, slová, ktoré sa nezmestia pohodlne do riadku, rozdelí. Od druhu dokumentu závisí, ako sú odstavce sádzané. Normálne je prvý riadok odstavca pevne stanovený a medzi dvoma odstavcami nie je žiadne voľné miesto. Viac informácií nájdete v kapitole 5.3.2.

V špeciálnych prípadoch môžeme prikázať \LaTeX u, aby zlomil riadok (ukončil ho) v danom mieste:

```
\ \ alebo \newline
```

začína nový riadok bez toho, aby začal nový odsek,

```
\ \*
```

dodatočne zakazuje zalomenie strany po vyplnení príkazu na zalomenie riadku.

```
\newpage
```

začína novú stranu.

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

robia presne to, čo hovoria ich názvy. Umožňujú autorovi ovplyvniť ich činnosť pomocou parametra n . Tento parameter môže nadobúdať hodnoty od 0 do 4. Nastavením parametra n na hodnotu nižšiu ako 4, nechávate \LaTeX u možnosť ignorovať váš príkaz v prípade, ak by výstup vyzeral veľmi zle. Nepleťte si tieto „break“ príkazy s príkazmi „new“. Hoci zadávate príkaz „break“ (ukončí), \LaTeX sa stále pokúša vyrovnáť pravý okraj strany a vytvorí stranu s požadovanou celkovou dĺžkou, ako to bude popísané

v nasledujúcom oddiele. Ak chcete skutočne začať „nový riadok“, použite odpovedajúci príkaz. Uhádnite jeho názov!

\LaTeX sa vždy pokúša dosiahnuť čo najlepšie zarovnávanie riadkov. Ak nevie nájsť spôsob, ako zarovnať riadky v súlade s jeho vysokými požiadavkami, nechá jeden riadok prečnievať za pravý okraj odstavca. Počas spracovania vstupného súboru sa potom \LaTeX sťažuje na („overfull hbox“). Toto sa najčastejšie stáva, keď \LaTeX nevie nájsť vhodné miesto na rozdelenie slova.² Pomocou príkazu `\sloppy` môžete \LaTeX u prikázať, aby trochu povolil v svojich požiadavkách. V takom prípade predchádza pretečeným riadkom zväčšením medzier medzi písmenami — aj keď konečný výstup nie je optimálny. Aj v takomto prípade vás \LaTeX varuje („underfull hbox“). Vo väčšine týchto prípadov nevyzerá výstup príliš dobre. Príkaz `\fussy` vráti \LaTeX na jeho štandardné správanie!

2.2.2 Delenie slov

\LaTeX rozdeľuje slová vždy, keď je to potrebné. V prípade, ak \TeX nevie na základe svojich algoritmov nájsť vhodné miesto na rozdelenie slova, môžete túto situáciu napraviť a nasledujúcim príkazom mu povedať o výnimke.

Príkaz

```
\hyphenation{zoznam slov}
```

spôsobí, že slová uvedené v parametroch, budú delené len na miestach označených „-“. Argument by by obsahovať len slová zložené z normálnych písmen alebo radšej znakov, ktoré sú považované za normálne písmená v aktívnom kontexte. Rozdeľovacie pravidlá sa ukladajú pre jazyk, ktorý je aktívny v momente keď sa príkaz objaví. To znamená, že ak umiestnite rozdeľovací príkaz do preambuly vášho dokumentu, ovplyvní anglické delenie slov. Ak príkaz umiestnite za `\begin{document}` a ak použijete nejaký balík pre národnú podporu ako `babel`, tak príkazy delenia budú aktívne v jazyku aktivovanom pomocou `babel`.

Nižšie uvedený príklad umožní rozdeliť slovo „rozdeľovanie“ rovnako ako „Rozdeľovanie“, ochráni slová „FORTRAN“, „Fortran“ a „fortran“ pred rozdelením v celom texte. Stav písmen (veľké—malé) je ignorovaný. V zadaní slov nie sú povolené žiadne špeciálne znaky alebo symboly.

Example:

```
\hyphenation{FORTRAN roz-de-lo-va-nie}
```

Príkaz `\-` umožňuje slovo v danom mieste rozdeliť. Len takéto miesto sa zároveň stáva miestom, kde môže byť dané slovo rozdelené. Tento prí-

²Hoci vás \LaTeX varuje, keď sa to stane (Overfull hbox), nie je vždy jednoduché nájsť takéto riadky. Ak v príkaze `\documentclass` použijete voľbu `draft`, budú tieto riadky označené hrubou čiernou čiarou na pravom okraji.

kaz je zvlášť užitočný pre slová, obsahujúce špeciálne znaky (napr. znaky s diakritikou), pretože L^AT_EX takéto slová automaticky nedelí.³

Myslím, že je to: su\ -per\ -cal\ -%
i\ -frag\ -i\ -lis\ -tic\ -ex\ -pi\ -%
al\ -i\ -do\ -cké

Myslím, že je to: supercalifragilisticexpialido-
cké

Viacero slov na jednom riadku môžeme udržať pomocou príkazu

`\mbox{text}`

Zabezpečí, že jeho obsah ostane v každom prípade spolu.

Budem mať nové telefónne číslo ---
`\mbox{0116 291 2319}`.

Budem mať nové telefónne číslo —
0116 291 2319.

Parameter
`\mbox{\emph{filename}}` musí
obsahovať názov súboru.

Parameter *filename* musí obsahovať názov sú-
boru.

2.3 Reľazce pripravené na použitie

V niektorých príkladoch na predchádzajúcich stranách ste videli veľmi jednoduché L^AT_EXovské príkazy na vytlačenie špeciálnych textových reľazcov:

Command	Example	Description
<code>\today</code>	2. mája 2001	Priebežný dátum v zapnutom jazyku
<code>\TeX</code>	T _E X	Meno vášho obľúbeného sadzača
<code>\LaTeX</code>	L ^A T _E X	The name of the Game
<code>\LaTeXe</code>	L ^A T _E X 2 _ε	Súčasná inkarnácia L ^A T _E Xu

2.4 Špeciálne znaky a symboly

2.4.1 Úvodzovky

Na vytvorenie úvodzoviek sa *nepoužíva* príkaz " ako na písacom stroji. Pri tlači existujú špeciálne otváracie a zatváracie úvodzovky. V L^AT_EXu sa používajú dva znaky ‘ na otvorenie a dva ’ na zatvorenie úvodzoviek.⁴

³Toto platí vo všeobecnosti. Napr. C_SL^AT_EX ich delí (pozn. prekl.).

⁴Pri použití štýlu `slovak.sty`, resp. `czech.sty`, môžete písať „naše“ úvodzovky pomocou príkazu `\uv{naše}` (pozn. prekl.)

‘‘Prosím, stlačte kláves ‘x’.’’

“Prosím, stlačte kláves ‘x’.”

2.4.2 Pomlčky a rozdeľovníky

\LaTeX pozná štyri druhy pomlčiek. Tri z nich dostanete pomocou rôzneho počtu po sebe idúcich pomlčiek. Štvrtý znak nie je vlastne ani pomlčka, je to matematický znak mínus:

X-násobný\\
strany 13--67\\
áno---alebo nie? \\
\$0\$, \$1\$ a \$-1\$

X-násobný
strany 13–67
áno—alebo nie?
0, 1 a –1

Názvy týchto pomlčiek sú: ‘-’ rozdeľovník (spojovník, hyphen), ‘-’ krátka pomlčka (en-dash), ‘—’ pomlčka (em-dash) a ‘-’ znak mínus.

2.4.3 Vlnka (~)

Znak, ktorý môžeme často vidieť ako súčasť web-adresy je vlnka. V \LaTeX u ju môžete vygenerovať použitím \sim , ale výsledok: \sim nie je v skutočnosti to, čo chcete. Namiesto toho skúste:

`http://www.rich.edu/~{bush} \\`
`http://www.clever.edu/\simdemo`

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

2.4.4 Bodky (...)

Na písacom stroji zaberá čiarka alebo bodka rovnaké miesto ako každý iný znak. Pri tlači kníh zaberajú tieto znaky len veľmi malú plochu a nachádzajú sa veľmi blízko predchádzajúceho písmena. Preto nemôžete napísať „bodky“ len napísaním troch bodiek, pretože medzery medzi nimi by boli zlé. Namiesto toho sa používa špeciálny príkaz, nazývaný

`\ldots`

Nie ako ... ale takto:\\
New York, Tokyo, Budapest, \ldots

Nie ako ... ale takto:
New York, Tokyo, Budapest, ...

2.4.5 Ligatúry

Niektoré písmena sa nesádzajú len tak, že sa vy tlačia ako rôzne písmená vedľa seba, ale niekedy sa vy tlačia s použitím špeciálnych znakov.

`ff fi fl ffi ...` namiesto `ff fi fl ffi ...`

Týmto takzvaným ligatúram môže byť zabránené vloženíím `\mbox{}` medzi písmená, ktorých sa to týka. Toto môže byť nevyhnutné pri zložených slovách.

Nie `shelfful`
ale `shelf\mbox{ }ful`

Nie `shelfful`
ale `shelfful`

2.4.6 Diakritika a špeciálne znaky

L^AT_EX podporuje užívanie diakritiky a špeciálnych znakov z mnohých jazykov. Tabuľka 2.1 ukazuje všetky druhy diakritiky na písmene o. Prirodzene na ostatných písmenách to funguje tiež.

Aby sa dal použiť niektorý znak na i alebo j, musia sa najprv odstrániť ich bodky. Toto je zabezpečené napísaním `\i` a `\j`.

`H\^otel, na\"i ve, \'el\'eve,\\`
`sm\o rrebr\o d, !'Se\~norita!,\\`
`Sch\"onbrunner Schlo\ss{}`
`Stra\ss e`

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

Tabuľka 2.1: Diakritika a špeciálne znaky

<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>ó</code>	<code>\.o</code>	<code>ö</code>	<code>\"o</code>	<code>ç</code>	<code>\c c</code>
<code>ř</code>	<code>\u o</code>	<code>ř</code>	<code>\v o</code>	<code>ř</code>	<code>\H o</code>	<code>q</code>	<code>\c o</code>
<code>ř</code>	<code>\d o</code>	<code>ř</code>	<code>\b o</code>	<code>ř</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>

2.5 Medzinárodná jazyková podpora

Ak chcete písať dokument v inom jazyku ako anglickom, existujú dve oblasti v ktorých musí byť \LaTeX správne nakonfigurovaný:

1. Všetky automaticky vytvárané textové reťazce⁵ musia byť prispôbené na nový jazyk. Pre mnohé jazyky môžu byť tieto zmeny vykonané použitím balíka `babel` od Johanna Braamsa.
2. \LaTeX potrebuje poznať pravidlá delenia slov nového jazyka. Dostať pravidlá delenia slov do \LaTeX u je trošku zložitejšie. Znamená to prebudovanie formátu so sprístupnením odlišných vzorov delenia slov. Váš *Local Guide* [4] by mal o tomto poskytnúť viac informácií.

Ak je už váš systém náležite nastavený, môžete aktivovať balík `babel` pridaním príkazu

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

nasledujúcim po príkaze `\documentclass`. V *Local Guide* [4] by malo byť uvedené, ktoré *jazyky* podporuje váš systém. Babel bude automaticky aktivovať vhodné pravidlá delenia slov jazyka, ktorý ste zvolili. Ak váš formát \LaTeX u (napr. `cslatex`) nepodporuje delenie slov vo vami zvolenom jazyku, babel bude aj tak funkčný, ale neumožní delenie, ktoré by malo úplne negatívny vplyv na vizuálny zjav vášho dokumentu.

Pre niektoré jazyky špecifikuje `babel` aj nové príkazy, zjednodušujúce vkladanie špeciálnych znakov. Napríklad v nemeckom jazyku existuje mnoho špeciálnych znakov (äöü). Pomocou `babelu` môžete napísať ö pomocou príkazu `"o` namiesto `\"o`.

Niektoré počítačové systémy umožňujú vkladanie špeciálnych znakov priamo z klávesnice. \LaTeX vie spracovať aj takéto znaky. Od decembrového vydania $\LaTeX 2_{\epsilon}$ v roku 1994, obsahuje rozšírovaná verzia $\LaTeX 2_{\epsilon}$ podporu rôznych vstupných kódovaní. Pozrite si balík `inputenc`. Počas používania tohoto balíku si musíte uvedomiť, že iní ľudia nemusia mať možnosť zobraziť si váš zdrojový súbor na svojom počítači, pretože môžu používať iné kódovanie. Napríklad „nemecký“ znak ä je na PC označený ako 132 a na niektorých UNIXovských systémoch, používajúcich ISO-LATIN 1, je označený ako 228. Preto používajte tieto nastavenia opatrne.

Iná vec je kódovanie fontu. Definuje, na ktorej pozícii \TeX ovského fontu je uložené každé písmeno. Originálne \TeX ovské fonty Computer Modern obsahujú len 128 znakov starej 7-bitovej ASCII tabuľky. Ak sú požadované akcentované znaky, vytvára ich \TeX kombinovaním normálnych písmen s akcentom. Hoci výsledný výstup môže vyzeráť bezchybne, tento prístup neumožňuje automatické delenie slov obsahujúcich písmená s diakritikou.

⁵Obsah, Zoznam obrázkov,

Našťastie súčasné distribúcie T_EXu obsahujú EC fonty. Tieto fonty vyzerajú ako fonty Computer Modern, ale obsahujú aj zvláštne znaky pre akcentované písmená, používané v európskych jazykoch. Použitím týchto fontov môžete skvalitniť delenie slov v neanglických dokumentoch. EC fonty sa sprístupňujú použitím balíka `fontenc` v preambule vášho dokumentu.

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

2.6 Medzery medzi slovami

Aby L^AT_EX dosiahol vo výstupe zarovnaný pravý okraj, vkladá medzi slová medzery rôznych veľkostí. Za vetou vkladá napatrne väčšiu medzeru, čo robí text čitateľnejším. L^AT_EX sa domnieva, že vety končia bodkou, otáznikom alebo výkričníkom. Ak nasleduje bodka za veľkým písmenom, nie je to považované za koniec vety, nakoľko je to jav typický pre skratky.

Každá výnimka z tohoto pravidla musí byť upresnená autorom. Opačné lomítko pred medzerou vytvára medzeru, ktorá nebude zväčšená. Znak vlnovky ‘~’ vytvára medzeru, ktorá nesmie byť zväčšená a navyše zabráňuje ukončeniu riadku v danom mieste. Príkaz `\@` pred bodkou určuje, že táto bodka ukončuje vetu, aj keď je pred ňou veľké písmeno.

```
Dr.~Smith bol rád, že ju videl\\
napr.~obr.~5\\
Mám rád BASIC\@. A čo vy?
```

```
Dr. Smith bol rád, že ju videl
napr. obr. 5
Mám rád BASIC. A čo vy?
```

Zväčšeniu medzery za bodkou môžete zabrániť použitím príkazu

```
\frenchspacing
```

ktorý prikáže L^AT_EXu, aby *nevkladal* o nič väčšie medzery za bodkou, ako za normálnym znakom. Toto je bežné pre neanglické jazyky, okrem bibliografií. Ak použijete `\frenchspacing`, príkaz `\@` nie je potrebný.⁶

2.7 Tituly, kapitoly a oddiely

Aby ste pomohli čitateľovi alebo čitateľke nájsť cestu vašim dokumentom, mali by ste ho rozdeliť na kapitoly, oddiely a pododdiely. Toto L^AT_EX zabezpečuje pomocou špeciálnych príkazov, ktoré používajú názov oddielu ako svoj argument. Je na vás, aby ste ich použili v správnom poradí.

⁶Ak používate voľbu `slovak`, resp. `czech`, je príkaz `\frenchspacing` automaticky aktívny (pozn. prekl.)

Pre štýl `article` sú dostupné nasledujúce druhy delenia:

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

Pre štýly `report` a `book` môžete navyše použiť nasledujúce dva príkazy:

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

Keďže `article` nepozná kapitoly, je úplne jednoduché pridávať články ako kapitoly do knihy. Medzery medzi oddielmi, číslovanie a veľkosť fontov ich názvov sú nastavené automaticky \LaTeX om.

Dva z uvedených príkazov na vytvorenie oddielov sú trochu špeciálne:

- Príkaz `\part` nemá vplyv na číselné poradie kapitol.
- Príkaz `\appendix` nemá parameter. Mení iba číselné označenie kapitol na písmená⁷.

\LaTeX vytvára obsah použitím názvov oddielov a im príslušných čísel strán, ktoré získal pri predchádzajúcom spracovaní textu. Príkaz

```
\tableofcontents
```

vytvorí obsah a umiestni ho na miesto, kde sa daný príkaz nachádza. Nový dokument musí byť spracovaný („pre \LaTeX ovaný“) dvakrát, aby sme dostali správny obsah. V niektorých prípadoch môže byť potrebné spracovať dokument tretíkrát. \LaTeX vám povie, kedy je to potrebné.

Všetky oddielové príkazy, uvedené vyššie, existujú aj v „hviezdičkovej“ forme. „Hviezdičková“ verzia príkazu sa vytvára pridaním znaku hviezdíčky `*` za názvom príkazu. Tieto príkazy vytvoria hlavičku oddielu, ktorý nebude číslovaný, a ani uvedený v obsahu. Napríklad namiesto príkazu `\section{Help}` napíšeme príkaz `\section*{Help}`.

Štandardne je názov oddielu vypísaný v obsahu presne tak, ako bol napísaný v texte. Niekedy to však nie je možné, pretože názov oddielu je príliš dlhý na to, aby sa do obsahu zmestil. Údaj pre obsah môže byť preto špecifikovaný ako nepovinný parameter pred aktuálnym nadpisom.

```
\chapter[Čítajte to! Je to napínavé]{Toto je veľmi dlhý
                                     a zvlášť nudný nadpis}
```

⁷Pri štýle `article` mení číslovanie oddielov

Titul celého dokumentu je vytváraný príkazom

```
\maketitle
```

Obsah titulu sa musí definovať príkazmi

```
\title{...}, \author{...} a prípadne \date{...}
```

predtým, ako je zavolaný príkaz `\maketitle`. V argumente `\authors` môžete vložiť viacero mien, oddelených príkazmi `\and`.

Príklad niektorých vyššie spomenutých príkazov môžete nájsť na obrázku 1.3 na strane 9.

Okrem oddielových príkazov, vysvetlených vyššie, $\text{\LaTeX} 2_{\epsilon}$ pozná navyše 3 príkazy, ktoré sa používajú v triede `book`:

```
\frontmatter, \mainmatter a \backmatter
```

Sú užitočné na rozdelenie vašej publikácie. Prerobia nadpisy kapitol a číslovanie strán tak, ako by ste to očakávali v knihách.

2.8 Krížové odkazy

V knihách, správach a článkoch sú často krížové odkazy na obrázky, tabuľky a špeciálne časti textu. \LaTeX poskytuje na vytváranie krížových odkazov nasledujúce príkazy

```
\label{značka}, \ref{značka} a \pageref{značka}
```

kde *značka* je identifikátor, zvolený autorom. \LaTeX nahrádza `\ref` číslom oddielu, pododdielu, obrázku, tabuľky, rovnice alebo vety, odpovedajúcim miestu, kde bol príkaz `\label` zadaný. `\pageref` vytlačí číslo strany, odpovedajúcej miestu, kde bol príkaz `\label` zadaný.⁸ Rovnako ako názvy oddielov sú aj čísla použité z predchádzajúceho behu \LaTeX u.

Odkaz k tomuto pododdielu
`\label{sec:tento}` vyzerá:
`\uv{pozri oddiel~\ref{sec:tento} na`
`strane~\pageref{sec:tento}.}`

Odkaz k tomuto pododdielu vyzerá: „pozri
oddiel 2.8 na strane 27.“

⁸Všimnite si, že tieto príkazy si nevedomujú, na čo odkazujú. `\label` len uloží posledné automaticky vygenerované číslo.

2.9 Poznámky pod čiarou

Pomocou príkazu

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

sa vytvorí na konci danej strany poznámka pod čiarou. Poznámky by mali byť vždy použité za slovom⁹ alebo za vetou, na ktorú sa vzťahujú.¹⁰

Poznámky pod čiarou\footnote{Toto je poznámka pod čiarou} sú často používané ľuďmi, používajúcimi \LaTeX.

Poznámky pod čiarou^a sú často používané ľuďmi, používajúcimi L^AT_EX.

^aToto je poznámka pod čiarou

2.10 Zvýraznené slová

V rukopise vytvorenom na písacom stroji sa dôležité slová zvýrazňujú obyčajne podčiarknutím. V tlačенých knihách sa však takéto slová *zvýrazňujú* vytlačením *kurzívou*. L^AT_EX na *zvýraznenie textu* poskytuje príkaz

```
\emph{text}
```

Jeho argumentom je text, ktorý chceme zvýrazniť. Čo príkaz v danom prípade urobí so svojim argumentom, závisí na kontexte:

```
\emph{Ak použijete
\emph{zvýrazňovanie} v už
zvýraznenom texte,
\LaTeX} použije
na zvýraznenie \emph{vzpriamený}
typ písma (\emph{antikvu}).}
```

Ak použijete zvýrazňovanie v už zvýraznenom texte, L^AT_EX použije na zvýraznenie vzpriamený typ písma (antikvu).

Všimnite si, prosím, rozdiel medzi príkázaním L^AT_EXu niečo *zvýraniť* a požiadanim ho použiť iný *font*:

```
\textit{Môžete tiež
\emph{zvýrazniť} text ak
je sádzaný kurzívou,}
\textsf{vo fonte
\emph{sans-serif},}
\texttt{alebo štýlom
\emph{typewriter}.}
```

Môžete tiež zvýrazniť text ak je sádzaný kurzívou, vo fonte sans-serif, alebo štýlom typewriter.

⁹Napríklad týmto.

¹⁰Poznámky, vzťahujúce sa ku vete alebo jej časti by preto mali byť uvedené za čiarkou alebo bodkou.

2.11 Prostredia

Aby sa dal napísať text rôzneho typu, \LaTeX má definovaných množstvo prostredí na viac rôznych druhov formátovania:

```
\begin{name} text \end{name}
```

Názov prostredia je *name*. Prostredia môžu byť volané viackrát (aj jedno v druhom), pokiaľ je dodržané ich poradie

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

V nasledujúcich podkapitolách sú opísané všetky dôležité prostredia.

2.11.1 Odrážky, vyratovania a popisy

Prostredie `itemize` je vhodné na vytváranie jednoduchých odrážok, prostredie `enumerate` na vytváranie číslovaných zoznamov a `description` je prostredie na vytváranie popisov.

```
\begin{enumerate}
\item Prostredia môžete
kombinovať podľa chuti:
\begin{itemize}
\item Ale môže to začať
vyzerať zle.
\item[-] S pomlčkou..
\end{itemize}
\item Preto si pamätajte:
\begin{description}
\item[Hlúposti] sa nestanú
chyt\~{r}\~{y}\~{m}i kvôli tomu,
že budú v\~{zozname}.
\item[Chytré] veci však, môžu
byť podané skvele v\~{zozname}.
\end{description}
\end{enumerate}
```

1. Prostredia môžete kombinovať podľa chuti:

- Ale môže to začať vyzerať zle.
- S pomlčkou..

2. Preto si pamätajte:

Hlúposti sa nestanú chytrými kvôli tomu, že budú v zozname.

Chytré veci však, môžu byť podané skvele v zozname.

2.11.2 Vlajočková sadzba a centrovanie

Prostredia `flushleft` a `flushright` vytvárajú odstavce, ktoré sú zarovnané buď doľava, alebo doprava. Prostredie `center` vytvára centrovany text. Ak nezadáte `\`, aby ste upresnili lámanie riadkov, \LaTeX ho určí sám.

```
\begin{flushleft}
Tento text je\\ zarovnaný doľava.
\LaTeX{} sa nepokúša vytvoriť
každý riadok rovnakej dĺžky.
\end{flushleft}
```

Tento text je
zarovnaný doľava. \LaTeX sa nepokúša
vytvoriť každý riadok rovnakej dĺžky.

```
\begin{flushright}
Tento text je zarovnaný\\ doprava.
\LaTeX{} sa nepokúša vytvoriť
každý riadok rovnakej dĺžky.
\end{flushright}
```

Tento text je zarovnaný
doprava. \LaTeX sa nepokúša vytvoriť každý
riadok rovnakej dĺžky.

```
\begin{center}
V~strede \\ Zeme
\end{center}
```

V strede
Zeme

2.11.3 Dlhšie citácie a verše

Prostredie `quote` je vhodné na citácie, dôležité frázy a príklady.

```
Typografické pravidlo palca
pre dĺžku riadku:
\begin{quote}
Žiaden riadok nemá obsahovať
viac ako 66~znakov.

Preto majú strany \LaTeX u~nastavené
ako východzie také veľké okraje.
\end{quote}
Preto sa v~tlačí novín často
používa viacero stĺpcov.
```

Typografické pravidlo palca pre dĺžku riadku:

Žiaden riadok nemá obsahovať
viac ako 66 znakov.

Preto majú strany \LaTeX u na-
stavené ako východzie také veľké
okraje.

Preto sa v tlači novín často používa viacero
stĺpcov.

Prostredia `quotation` a `verse` sú podobné. Prostredie `quotation` sa využíva na dlhšie citácie, skladajúce sa z viacerých odstavcov, pretože vie odsadzovať odstavce. Prostredie `verse` je užitočné na písanie básní, kde je potrebné lámať riadky. Riadky sú delené pomocou príkazu `\` použitého na konci riadku, alebo pomocou voľného riadku za každým veršom.

Naspamäť poznám len jednu anglickú báseň. Je o Humpty Dumpty.

```
\begin{flushleft}
\begin{verse}
Humpty Dumpty sat on a wall:\\
Humpty Dumpty had a great fall.\\
All the King's horses and all
the King's men\\
Couldn't put Humpty together
again.
\end{verse}
\end{flushleft}
```

Naspamäť poznám len jednu anglickú báseň. Je o Humpty Dumpty.

```
Humpty Dumpty sat on a wall:
Humpty Dumpty had a great
fall.
All the King's horses and all
the King's men
Couldn't put Humpty together
again.
```

2.11.4 Doslovná tlač

Text, uzavretý medzi `\begin{verbatim}` a `\end{verbatim}`, bude priamo vytlačený tak, ako by bol napísaný na písacom stroji, so všetkými zakončeniami riadkov a medzerami, bez uskutočnenia akéhokoľvek \LaTeX ového príkazu.

Vo vnútri odstavca, môžete dosiahnuť rovnaký výsledok aj pomocou príkazu

```
\verb+text+
```

+ je len oddeľovací znak. Môžete použiť ľubovoľný znak okrem písmen, znaku * a medzier. Množstvo \LaTeX ových príkladov v tejto brožúre je vysádzaných pomocou tohto príkazu.

Príkaz `\verb|\ldots| \ldots`

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

Príkaz `\ldots ...`

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

```
\begin{verbatim*}
verzia prostredia
      verbatim
s~hviezdičkou zvýrazňuje
medzery   v~texte
\end{verbatim*}
```

```
verzia_prostredia
UUUUUUverbatim
s~hviezdičkou_zvýrazňuje
medzeryUUUUv~texte
```

Príkaz `\verb` sa môže použiť v podobnom tvare s hviezdíčkou:

```
\verb*|napríklad takto :-)|
```

```
napríklad_takto_:-)_
```

Prostredie `verbatim` a príkaz `\verb` sa nesmú používať v parametroch iných príkazov.

2.11.5 Vytváranie tabuliek

Prostredie `tabular`¹¹ sa používa na sadzbu krásnych tabuliek s voliteľnými vertikálnymi a horizontálnymi čiarami. \LaTeX určuje šírku stĺpcov automaticky.

Argument *typ stĺpca* príkazu

```
\begin{tabular}{typ stĺpca}
```

definuje formát tabuľky. Parameter `l` vytvorí stĺpec zarovnaný doľava, `r` doprava, `c` centrováný stĺpec, parameter `p{šírka}` vytvorí stĺpec danej šírky (zarovnaný doľava) a nakoniec príkaz `|` vytvorí zvislú čiaru.¹²

V prostredí `tabular` sa pomocou znaku `&` prepínate do nasledujúceho stĺpca, `\\` začína nový riadok, `\hline` vkladá horizontálnu čiaru a `\cline` podčiarkne zvolené stĺpce.

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

7C0	hexadecimal
3700	octal
11111000000	binary
<hr/>	
1984	decimal

```
\begin{tabular}{|p{4.7cm}|}
\hline
Vítajte v Boxyho odstavci.
Úprimne dúfame, že sa
Vám bude naša show páčiť.
\\ \hline
\end{tabular}
```

Vítajte v Boxyho odstavci. Úprimne dúfame, že sa Vám bude naša show páčiť.
--

¹¹Tabuľky, vytvorené v prostredí `tabular` nemôžu byť rozdelené na dve strany. Na dlhé tabuľky sa používajú štýly `supertab.sty` a `longtbl.sty` (pozn. prekl.)

¹²Balík `hhline.sty` umožňuje rôzne spôsoby, akými sa majú čiary v tabuľkách pretínať (pozn. prekl.)

Pomocou príkazu `@{...}` je možné špecifikovať oddelovač stĺpcov. Tento príkaz ruší medzistĺpcové medzery a nahrádza ich tým, čo sa nachádza v zložených zátvorkách. Jeden zo spôsobov použitia tohoto príkazu bude uvedený nižšie pri probléme zarovnania podľa desatinnej bodky. Iný možný spôsob využitia je potlačenie okrajových medzier v tabuľke pomocou príkazu `@{}`.

```
\begin{tabular}{@{} l @{}}
\hline
žiadne okrajové medzery\\ \hline
\end{tabular}
```

žiadne okrajové medzery

```
\begin{tabular}{l}
\hline
medzery napravo a naľavo\\
\hline
\end{tabular}
```

medzery napravo a naľavo

Keďže nie je zabudovaný spôsob na zarovnávanie číselných stĺpcov podľa desatinnej bodky¹³, môžeme trochu zašvindľovať a použiť dva stĺpce: doprava zarovnaný pre celé čísla a doľava zarovnaný pre desatinnú časť. Príkaz `@{.}`, použitý v riadku `\begin{tabular}`, nahrádza bežnú medzistĺpcovú medzeru znakom „.“, vytvárajúc tak dojem, že daný stĺpček je zarovnaný podľa desatinnej bodky. Nezabudnite nahradiť desatinnú bodku vo vašich číslach oddelovačom stĺpcov (`&`)! Označenie stĺpca môžeme nad náš „číselný stĺpec“ umiestniť pomocou príkazu `\multicolumn`, ktorým sa spája viac stĺpcov.

```
\begin{tabular}{c r @{.} l}
Výraz Pí & & 
\multicolumn{2}{c}{Hodnota} \\
\hline
$\pi$ & 3&1416 & \\
$\pi^{\pi}$ & 36&46 & \\
$(\pi^{\pi})^{\pi}$ & 80662&7 & \\
\end{tabular}
```

Výraz Pí	Hodnota
π	3.1416
π^{π}	36.46
$(\pi^{\pi})^{\pi}$	80662.7

```
\begin{tabular}{|c|c|}
\hline
\multicolumn{2}{|c|}{\textbf{V strede}} \\
\hline
Vľavo & Vpravo \\
\hline
\end{tabular}
```

V strede	
Vľavo	Vpravo

¹³Ak máte inštalovaný balík `tools`, mrknite sa na balík `dcolumn`

2.11.6 Nastavenie tabulátorov

Prostredie `tabbing`¹⁴ umožňuje autorovi pracovať s tabulátormi. Presnejší popis môžete nájsť napríklad v [12].

Na rozdiel od prostredia `tabular`, v prostredí `tabbing` sa môže objaviť koniec stránky, teda jeho dĺžka je neobmedzená. Využívanie tabulátorov je možné používať napríklad na formátovanie zápisov štruktúrovaných programov. Nasleduje stručný popis jednotlivých príkazov.

- `\=` – nastavenie tabulačnej zarážky
- `\>` – prechod na ďalšiu zarážku
- `\<` – prechod na predchádzajúcu zarážku
- `\` – nový riadok
- `\kill` – ukončenie „vzorového“ riadku
- `\+` – posunutie ľavého okraja textu na ďalšiu zarážku
- `\-` – posunutie ľavého okraja textu o zarážku doľava
- `\'` – umiestnenie pravého okraja textu na pravý okraj predchádzajúcej zarážky (alebo `1\tabbingsep` pred aktuálnu zarážku)
- `\‘` – posunie nasledujúci text na pravý okraj riadku
- `\pushtabs` – uloženie pozícií všetkých zarážok
- `\poptabs` – vyvolanie pozícií zarážok
- `\a ...` – v prostredí `tabbing` nefungujú príkazy `\=`, `\'`, `\‘`, vytvárajúce akcenty. Na tieto akcenty v prostredí `tabbing` používame príkazy `\a=`, `\a'`, `\a‘`

Nasledujúci príklad ilustruje použitie „vzorového“ riadku a všetkých vyššie uvedených príkazov, s výnimkou `\pushtabs` a `\poptabs`. `TeX` sa nastaví na zarážku nezávisle od toho, či sa text bude prekrývať, alebo nie.

				<code>\begin{tabbing}</code>	
				<code>raz \=dva \quad \=tri\kill</code>	
raz	dva	tri	štyri	<code>raz \>dva \>tri \=štyri \+\</code>	
	dva			<code>dva\+\</code>	
		tri		<code>tri\+\</code>	
			štyri	<code>štyri\</code>	
		tri		<code>\<tri\</code>	
raz			dva	<code>\<\<\<raz \‘dva \</code>	
		tri		<code>\<tri\’\</code>	
			štyrikrát	<code>štyrikr\a’at\-\-\</code>	
	dva			<code>dva\-\</code>	
štyrtri	dva	raz		<code>štyri\>tri\>dva\>raz\</code>	
				<code>\end{tabbing}</code>	

¹⁴Pridané pri slovenskom preklade (pozn. prekl.)

2.12 Plávajúce objekty

V súčasnosti obsahuje väčšina publikácií množstvo obrázkov a tabuliek. Tieto elementy si vyžadujú zvláštny prístup, pretože sa nemôžu rozdeľovať na viacero strán. Jedným zo spôsobov riešenia tohoto problému je začať novú stranu zakaždým, keď je obrázok alebo tabuľka príliš veľká, aby sa zmestila na aktuálnu stranu. Tento prístup by zanechal strany čiastočne prázdne, čo v konečnom dôsledku vyzerá veľmi zle.

Riešením tohoto problému je „plávanie“ obrázku alebo tabuľky, ktorá nebude umiestnená na danej strane, ale na niektorej z nasledujúcich strán, kým daná strana bude zaplnená textom. \LaTeX poskytuje dve prostredia na výrobu plávajúcich objektov. Jedno pre tabuľky a jedno pre obrázky. Aby ste mohli plne využiť výhody týchto dvoch prostredí, je dôležité, aby ste približne chápali, ako s nimi \LaTeX interne narába. V opačnom prípade môžu byť tieto plávajúce objekty hlavným zdrojom frustrácie, pretože \LaTeX ich nikdy nedá tam, kde by ste ich chceli mať.

Pozrime sa na príkazy, ktoré nám \LaTeX ponúka pre plávajúce útvary:

Všetok materiál, umiestnený v prostrediach `figure` a `table`, je považovaný za plávajúci. Obe tieto plávajúce prostredia podporujú nepovinný parameter

```
\begin{figure}[špecifikátor umiestnenia]
```

alebo

```
\begin{table}[špecifikátor umiestnenia]
```

nazývaný *špecifikátor umiestnenia*. Tento parameter hovorí \LaTeX u, kam môže umiestniť daný plávajúci útvar. *Špecifikátor umiestnenia* je tvorený reťazcom *podmienok na umiestnenie plávajúceho objektu*. Pozri tabuľku 2.2.

Tabuľka môže začínať napríklad nasledujúcim riadkom

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

Špecifikátor umiestnenia `[!hbp]` povoľuje \LaTeX u umiestniť tabuľku priamo tu (**h**), alebo na spodok nejakej strany (**b**), alebo na špeciálnu stranu pre plávajúce objekty (**p**), a to aj v prípade, že to nebude vyzeráť veľmi dobre (!). Pokiaľ nie je zadaný žiaden špecifikátor, \LaTeX pracuje s `[tbp]`.

\LaTeX umiestni každý plávajúci objekt podľa špecifikátora umiestnenia, zadaného autorom. Ak plávajúci objekt nemôže byť umiestnený na aktuálnej strane, je zaradený do poradovníka *obrázkov* alebo *tabuliek*¹⁵. Keď sa začne nová strana, \LaTeX najprv skontroluje, či nie je možné vytvoriť špeciálnu stranu, zaplnenú len plávajúcimi objektami z poradovníkov. Ak to

¹⁵tieto sú FIFO – ‘first in, first out’ – zásobníky

nie je možné, \LaTeX vezme prvý objekt z poradovníka, ako keby bol len teraz umiestnený do textu a pokúša sa ho znovu umiestniť na základe špecifikátorov umiestnenia, ktoré boli pri ňom zadané (okrem „h“, ktoré už nie je prijateľné). Všetky ďalšie plávajúce objekty, ktoré sa v texte objavia, sú umiestnené do príslušného poradovníka. \LaTeX prísne dodržiava pôvodné poradie vzhľadom na každý druh plávajúcich objektov. To je príčinou toho, že obrázok, ktorý nemôže byť umiestnený, vytlačí všetky ďalšie obrázky na koniec dokumentu. Preto:

Ak \LaTeX neumiestňuje plávajúce objekty podľa vašich očakávaní, je to často jeden plávajúci objekt, ktorý upcháva jeden z dvoch poradovníkov plávajúcich objektov.

Keď sme si už vysvetlili náročnejšiu časť, existuje ešte pár vecí, ktoré by sme mali o prostrediach `table` a `figure` spomenúť. Príkazom

```
\caption{názov plávajúceho objektu}
```

môžete definovať názov plávajúceho objektu. Priebežné poradové číslo a označenie „Obrázok“ alebo „Tabuľka“ doplní \LaTeX .

Dva príkazy

```
\listoffigures a \listoftables
```

pracujú podobne ako príkaz `\tableofcontents`. Vytlačia zoznam obrázkov, respektíve tabuliek. V týchto zoznamoch je spomenutý celý názov daných objektov. Ak ste náchylní k používaniu dlhých názvov obrázkov (tabuliek),

Tabuľka 2.2: Podmienky na umiestnenie plávajúceho objektu

Špec.	Povolenie na umiestnenie plávajúceho objektu
h	<i>here (tu)</i> práve na tom mieste textu, kde bol daný príkaz vydaný. Toto je užitočné najmä pre malé plávajúce objekty.
t	na vrchu strany (<i>top</i>)
b	na spodku strany (<i>bottom</i>)
p	na zvláštnej strane, obsahujúcej len plávajúce objekty. (<i>page</i>)
!	bez ohľadu na vnútorné parametre ^a , ktoré môžu zabrániť umiestneniu plávajúceho objektu.

^aAko napríklad maximálny počet plávajúcich objektov na jednej strane

musíte mať kratšiu verziu názvu, ktorá pôjde do zoznamu. Toto sa zabezpečuje napísaním kratšej verzie názvu do hranatých zátvoriek za príkazom `\caption`.

```
\caption[Krátky]{DDD11111hhhhýýýý}
```

Príkazmi `\label` a `\ref` môžete vytvoriť odkazy na plávajúce objekty vo vnútri textu.

Nasledujúci príklad kreslí štvorec a vkladá ho do dokumentu. Toto môžete využiť, ak chcete nechať miesto na obrázky, ktoré chcete vložiť (vlepiť) do dokončeného dokumentu.

```
Obrázok~\ref{biela} je príkladom Pop-Art\,u.
\begin{figure}[!hbp]
\makebox[\textwidth]{\framebox[5cm]{\rule{0pt}{5cm}}}
\caption{Päťkrát päť centimetrov} \label{biela}
\end{figure}
```

Vo vyššie uvedenom príklade¹⁶ sa \LaTeX *skutočne ťažko* (!) pokúša umiestniť obrázok priamo *tu* (h). Ak to nie je možné, pokúša sa umiestniť obrázok na *spodku* (b) strany. Neuspejúc pri pokuse umiestniť obrázok na danej strane, zisťuje, či by nebolo možné vytvoriť stranu plávajúcich objektov, obsahujúcu tento obrázok a možno niekoľko tabuliek z poradovníka tabuliek. Ak nemá dostatok materiálu na stranu plávajúcich objektov, \LaTeX začne novú stranu a ešte raz bude narábať s obrázkom, ako keby sa práve teraz objavil v texte.

Za určitých podmienok môže byť nevyhnutné použitie príkazu

```
\clearpage alebo dokonca \cleardoublepage
```

Tento príkazuje \LaTeX okamžite umiestniť zvyšné plávajúce objekty nachádzajúce sa v poradovníkoch a až potom začať novú stranu. Na začiatok novej pravej strany nás nastaví príkaz `\cleardoublepage`.

Neskôr, v tomto úvode do \LaTeX , sa dozvieme, ako vkladáť PostScriptové obrázky do vašich \LaTeX ových dokumentov.

¹⁶za predpokladu, že poradovník obrázkov je prázdny

Kapitola 3

Sádzanie matematických vzorcov

Teraz ste pripravení! V tejto kapitole zaútočíme na najsilnejšie miesto T_EXu: sádzanie matematiky. Ale varujem vás, táto kapitola iba pohladká povrch. Keďže tu opísané veci postačujú väčšine ľudí, nezúfajte, keď nebudete vedieť nájsť riešenie vašich potrieb matematickej sadzby. Je vysoko pravdepodobné, že váš problém je vyriešený v AMS- \LaTeX u¹ alebo v niektorom inom balíku.

3.1 Všeobecné

\LaTeX má zvláštny režim na sádzanie matematiky. Matematický text je v odstavci písaný medzi $\text{\begin{math}}$ a $\text{\end{math}}$, medzi $\text{\$}$ a $\text{\$}$, alebo medzi \(a \) .

Sčítajte $\text{\$a}$ na druhú a $\text{\$b}$ na druhú tak, aby ste dostali $\text{\$c}$ na druhú. Alebo, použijúc matematickejší prístup:
$$\text{\$c}^{\text{\{2}}}=a^{\text{\{2}}+b^{\text{\{2}}}$$

Sčítajte a na druhú a b na druhú tak, aby ste dostali c na druhú. Alebo, použijúc matematickejší prístup:
$$c^2 = a^2 + b^2$$

\TeX sa vyslovuje ako \tau\epsilon\chi .
 $\text{\$100m}^{\text{\{3}}}$ vody
Toto vychádza z môjho \heartsuit .

\TeX sa vyslovuje ako $\tau\epsilon\chi$.
 100 m^3 vody
Toto vychádza z môjho \heartsuit .

Väčšie matematické vzorce alebo rovnice je výhodnejšie sádzať ako samostatné výrazy a nie ich len písať na samostatných riadkoch. Preto ich

¹CTAN:/tex-archive/macros/latex/packages/amslatex

musíte uzatvárať medzi `\[a \]` alebo medzi príkazy `\begin{displaymath}` a `\end{displaymath}`. Tieto príkazy vytvoria vzorce, ktoré nie sú číslované. Ak chcete, aby ich L^AT_EX čísloval, môžete použiť prostredie `equation`.

Sčítajte `a` na druhú a `b` na druhú tak, aby ste dostali `c` na druhú. Alebo, použijúc matematickejší prístup:

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

A práve jeden ďalší riadok.

Sčítajte a na druhú a b na druhú tak, aby ste dostali c na druhú. Alebo, použijúc matematickejší prístup:

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

A práve jeden ďalší riadok.

Pomocou `\label` a `\ref` sa v texte môžete odvolávať na rovnice.

```
\begin{equation} \label{eq:eps}
\epsilon > 0
\end{equation}
Z(\ref{eq:eps}) sme dostali
\ldots
```

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

Z (3.1) sme dostali ...

Zapamätajte si, že výrazy budú vytlačené rôznym spôsobom, ak budú rôzne zadané:

```
\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}
```

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

Medzi *matematickým režimom* a *textovým režimom* sú rozdiely. Napríklad v *matematickom režime*:

1. Väčšina medzier a ukončení riadkov nemá význam, všetky medzery sú buď odvodené z logického obsahu matematických výrazov, alebo musia byť zadané pomocou príkazov ako `\,`, `\quad` alebo `\qquad`.
2. Prázdne riadky nie sú dovolené. Iba jeden odstavec na jeden výraz.
3. Každé písmeno sa považuje za názov premennej a je vysádzané ako premenná. Ak chcete vo výraze použiť normálny text (normálny vzpriamený font a normálne vzdialenosti písmen), potom musíte použiť na napísanie textu príkaz `\text{rm}{...}`.

```
\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 \forall x \in \mathbf{R}
\end{equation}
```

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

Matematici môžu byť veľmi úzkostliví na symboly, ktoré sa používajú: konvenčne by sa tu mal používať font „blackboard bold“, ktorý je dosiahnuteľný príkazom `\mathbb` z balíka `amssymb` alebo `amssymb`. Posledný príklad by vyzeral

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

$$x^2 \geq 0 \quad \text{for all } x \in \mathbb{R}$$

3.2 Zoskupovanie v matematickom režime

Väčšina príkazov matematického režimu funguje len na nasledujúcom znaku. Teda ak chcete, aby sa daný príkaz týkal viacerých po sebe idúcich znakov, musíte ich uzavrieť do zložených zátvoriek: `{...}`.

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

$$a^x + y \neq a^{x+y} \quad (3.4)$$

3.3 Sádzanie matematických výrazov

V tejto časti budú opísané najdôležitejšie príkazy. Kompletný zoznam všetkých matematických znakov je v odseku 3.10 na strane 52.

Malé grécke písmená sa zadávajú ako `\alpha`, `\beta`, `\gamma`, ..., veľké písmená² sa zadávajú ako `\Gamma`, `\Delta`, ...

²V \LaTeX u nie je definovaná veľká Alfa, pretože vyzerá rovnako ako normálne A. Ak dôjde niekedy k zmenám v matematickom kódovaní, zmení sa to.

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

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

Exponenty a dolné indexy môžu byť zadané použitím znakov $\hat{\ } a _$.

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

`$e^{x^2} \neq e^{x^2}$`

`$a^{\{3\}_{ij}}$`

`$e^{\{x^2\}} \neq \{e^x\}^2$`

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

Druhá odmocnina sa zadáva príkazom `\sqrt`, a n -tá odmocnina pomocou `\sqrt[n]`. Veľkosť znaku odmocniny vypočíta L^AT_EX automaticky. Ak potrebujete len znak odmocniny, je potrebné použiť príkaz `\surd`.

`\sqrt{x}` \quad

`\sqrt{x^2 + \sqrt{y}}` \quad

`\sqrt[3]{2}` \quad `\sqrt[3]{2}` \quad `\sqrt[3]{2}`

`\surd[x^2 + y^2]`

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

Príkazy `\overline` a `\underline` vytvoria **horizontálne čiary** priamo nad alebo pod výrazom.

`\overline{m+n}`

$\overline{m+n}$

Príkazy `\overbrace` a `\underbrace` vytvoria dlhé **horizontálne svorky** nad alebo pod výrazom.

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

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

Aby ste mohli pridať k premenným akcenty, resp. značky ako, napríklad, malé šípky, vlnovky atď., môžete použiť príkazy, uvedené v tabuľke 3.1. Dlhé striešky a vlnovky, pokrývajúce viacero znakov, sa generujú pomocou príkazov `\widetilde` a `\widehat`. Symbol ' dáva čiarku (prime).

`\begin{displaymath}`

`y=x^2 \quad y^{\prime}=2x \quad y''=2`

`\end{displaymath}`

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

Vektory sú často označované pridaním malého znaku šípky nad premennú. Toto sa robí pomocou príkazu `\vec`. Na vyznačenie vektora z bodu A do bodu B sú užitočné dva príkazy `\overrightarrow` a `\overleftarrow`.


```
\begin{displaymath}
\vec a\quad\overrightarrow{AB}
\end{displaymath}
```

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

Názvy funkcií (napr. logaritmických) sa často sádzajú zvislým fontom (antikvou) a nie šikmým (kurzívou), ako premenné. Preto L^AT_EX obsahuje nasledujúce príkazy na sádzanie názvov najdôležitejších funkcií:

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

```
[\lim_{x \rightarrow 0}
\frac{\sin x}{x}=1]
```

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

Pre funkciu modulo existujú dva príkazy: `\bmod` pre binárny operátor „ a mod b “ a `\pmod` pre výrazy ako napríklad „ $x \equiv a \pmod{b}$ “.

Zlomky sa sádzajú pomocou príkazu `\frac{...}{...}`. Často sa uprednostňuje tvar zlomkov ako $1/2$ (najmä v textovom režime), pretože to vyzerá lepšie, ak zlomky neobsahujú veľké výrazy.

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

$1\frac{1}{2}$ hodiny

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

Na napísanie binomických koeficientov, alebo podobných štruktúr, môžete použiť príkaz `{... \choose ...}` alebo `{... \atop ...}`. Príkaz, spomínaný ako druhý, vytvorí rovnaký výstup ako prvý, avšak bez zátvoriek. (Upozorníme, že použitie týchto príkazov starého štýlu je výslovne zakázané v balíku `amsmath`. Sú nahradené príkazmi `\binom` a `\genfrac`. Ten druhý zahŕňa všetky odpovedajúce konštrukcie, napríklad môžete dostať pomocou `\newcommand{\newatop}[2]{\genfrac{}{}{0pt}{1}{#1}{#2}}` niečo podobné ako `\atop`.)

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

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

Pre binárne môže byť užitočné umiestnenie symbolov jeden nad druhým. `\stackrel{!}{=}` umiestni symbol daný ako prvý argument vo veľkosti indexu nad druhý symbol, ktorý bude stáť na svojej obvyklej pozícii.

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

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

Znak integrálu sa vytvára pomocou `\int` a **znak sumy** pomocou príkazu `\sum`. Horné a dolné limity sa zadávajú pomocou `^` a `_` ako pre dolný a horný index.³⁴

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

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

Pre **zátvorky** a ostatné oddeľovače existujú v `TeXu` všetky typy znakov (napr. `[< || ↓`). Okrúhle a hranaté zátvorky môžu byť priamo zadané z klávesnice pomocou príslušných klávesov, zložené zátvorky pomocou príkazu `\{`, všetky ostatné oddeľovače sú vytvárané pomocou špeciálnych príkazov (napr. príkazu `\updownarrow`). Zoznam všetkých oddeľovačov je v tabuľke 3.8 na strane 54.

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

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

Pokiaľ zadáte pred otvárací oddeľovač príkaz `\left` a pred uzatvárací oddeľovač príkaz `\right`, `TeX` automaticky vypočíta správnu veľkosť oddeľovača. Zapamätajte si, že každé `\left` musíte uzatvoriť príslušným `\right`. Ak nechcete napravo nič, použite neviditeľné „`\right.`“!

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

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

³ `AMS-LATEX` má navyše viacriadkové horné a dolné indexy

⁴ Príkazy `\textstyle`, `\displaystyle` a `\limits` menia spôsob ich umiestnenia. (pozn. prekl.)

Všimnime si, že ‘d’ v diferenciále sa zvykne písať antikvou – typom roman.

$\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ poskytuje ďalšie cesty na jemnú úpravu medzier medzi znakmi integrovania, konkrétne príkazy `\iint`, `\iiint`, `\iiiint`, a `\idotsint`. Pri natahnutom balíku `amsmath` môže byť vyššie uvedený príklad vysádzaný nasledujúcim spôsobom:

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

$$\iint_D dx dy$$

Kvôli ďalším podrobnostiam pozri elektronický dokument `testmath.tex` (rozširovaný spolu s $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}\text{om}$) alebo Kapitolu 8 *The L^AT_EX Companion* [3].

3.5 Vertikálne zarovnaný materiál

Na vysádzanie **matic** sa používa prostredie `array`. Pracuje podobne ako prostredie `tabular`. Príkaz `\\` sa využíva na ukončenie riadkov.

```
\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}
```

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

Prostredie `array` sa môže používať aj na sádzanie výrazov, ktoré majú jeden veľký oddeľovač pomocou znaku bodky („.“) ako neviditeľného `\right` oddeľovača:

```
\begin{displaymath}
y = \left\{ \begin{array}{ll}
a & \text{\texttrm{ak } $d > c$} \\
b+x & \text{\texttrm{ráno}} \\
l & \text{\texttrm{počas celého dňa}}
\end{array} \right.
\end{displaymath}
```

$$y = \begin{cases} a & \text{ak } d > c \\ b + x & \text{ráno} \\ l & \text{počas celého dňa} \end{cases}$$

Rovnako ako vnútri prostredia `tabular` môžete kresliť čiary aj v prostredí `array`, napr. na oddelenie prvkov matice:

```

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

```

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

Na sádzanie vzorcov, ktoré dĺžkou prekračujú hranicu jedného riadku, alebo na sádzanie sústav rovníc, môžete namiesto prostredia `equation` použiť prostredia `eqnarray` a `eqnarray*`. V prostredí `eqnarray` je každý riadok automaticky číslovaný. V prostredí `eqnarray*` riadky číslované nie sú.

Prostredia `eqnarray` a `eqnarray*` pracujú ako 3-stĺpcová tabuľka tvaru `{rcl}`, kde sa stredný stĺpec používa pre znak rovnosti, alebo nerovnosti, alebo iný znak, ktorý budete považovať za vhodný. Príkaz `\` ukončuje („láme“) riadky.

```

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

```

$$f(x) = \cos x \quad (3.5)$$

$$f'(x) = -\sin x \quad (3.6)$$

$$\int_0^x f(y)dy = \sin x \quad (3.7)$$

Určite si všimnete, že na oboch stranách stredného stĺpca, znaku rovnosti, je veľa voľného miesta. Ako uvidíte v nasledujúcom príklade, toto môže byť zmenšené príkazom `\setlength\arraycolsep{2pt}`.

Dlhé rovnice nie sú automaticky rozdeľované na pekné časti. Autor musí sám určiť, kde sa majú rozdeliť a nakoľko ich treba odsadiť. Na dosiahnutie tohoto sa najčastejšie používajú nasledujúce dve metódy:

```

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

```

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots \quad (3.8)$$

```

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

```

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \cdots \quad (3.9)$$

Príkaz `\nonumber` spôsobí, že \LaTeX danú rovnicu neočísluje.

Môže byť náročné vytvoriť pomocou týchto metód vertikálne zarovnané rovnice, ktoré vyzerajú dobre: balík `amsmath` poskytuje viac možností na riešenie tohoto problému (pozri prostredia `split` a `align`).

3.6 Duch

Nemôžeme vidieť duchov (fantómov), ale tieto ešte stále obývajú určité miesto v mysliach mnohých ľudí. \LaTeX nie je iný. Túto okolnosť môžeme použiť na niektoré zaujímavé triky s medzerami.

Pri vertikálnom zarovnávaní textu pomocou `^` a `_` môže byť niekedy \LaTeX tiež trochu osožný. Použitím príkazu `\phantom` môžete vymedziť priestor na písmená, ktoré nebudú v konečnom výstupe ukázané. Najlepšie to je vidieť na nasledujúcich príkladoch.

```

\begin{displaymath}
{}^{12}_6\text{C} \phantom{{}_6\text{C}}
\quad \text{as opposed to} \quad
{}^{12}_6\text{C}
\end{displaymath}

```

$${}^{12}_6\text{C} \quad \text{as opposed to} \quad {}^{12}_6\text{C}$$

```

\begin{displaymath}
\Gamma_{ij}^k \phantom{\Gamma_{ij}^k}
\quad \text{as opposed to} \quad
\Gamma_{ij}^k
\end{displaymath}

```

$$\Gamma_{ij}^k \quad \text{as opposed to} \quad \Gamma_{ij}^k$$

3.7 Veľkosť matematického fonu

V matematickom režime volí \TeX veľkosť fonu v závislosti od kontextu. Napríklad horné indexy sú sádzané menším fontom. Ak by ste chceli napísať v rovnici text fontom roman a použili by ste príkaz `\textrm`, mechanizmus na zmenu veľkosti fonu by nefungoval pretože príkaz `\textrm` dočasne prešiel do textového režimu. Namiesto príkazu `\textrm` musíte použiť príkaz `\mathrm`, aby mechanizmus na menenie veľkostí fontov fungoval aj naďalej.

Ale dávajte si pozor, `\mathrm` funguje dobre len na malých úsekoch. Medzery stále nie sú považované za znaky, negunguje ani diakritika.⁶

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

$$2^{\text{nd}} \quad 2^{\mathrm{nd}} \quad (3.10)$$

Niekedy napriek tomu, že \LaTeX vie vypočítať správnu veľkosť fontov, mu potrebujete zadať správnu veľkosť. V matematickom režime sa veľkosť písma zadáva pomocou štyroch príkazov:

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

Menenie štýlov ovplyvňuje tiež spôsob, akým sa zobrazujú hranice.

```
\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}
```

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

Toto je jeden z tých príkladov, kedy potrebujeme väčšie hranaté zátvorky ako nám poskytujú príkazy `\left[` `\right]`.

3.8 Vety, zákony, ...

Ak budete písať matematické dokumenty, pravdepodobne budete potrebovať spôsob, ako napísať „lemy“, „definície“, „axiómy“ a podobné štruktúry. \LaTeX toto zabezpečuje príkazom

```
\newtheorem{meno}[čítač]{text}[section]
```

Argument *meno* je krátke kľúčové slovo, ktoré sa využíva na označenie „teorémy“ v texte. Pomocou argumentu *text* definujete názov „teorémy“, ktorý sa objaví vo výslednom dokumente.

⁶Balík AMS- \LaTeX pri práci s príkazom `\textrm` rieši tieto problémy fontov.

Argumenty v hranatých zátvorkách sú nepovinné. Oba sa používajú na bližšie určenie číslovania, použitého pri danej „teoréme“. Argumentom *čítač* môžete bližšie určiť *meno* skôr uvedenej „teorémy“. Nová „teoréma“ tak bude číslovaná v nadväznosti na skôr uvedenú „teorému“. Argument *section* vám dovoľuje určiť oddiel, v ktorom chcete, aby bola vaša „teoréma“ číslovaná.

Po zadaní príkazu `\newtheorem` do hlavičky vášho dokumentu, môžete vo vašom dokumente použiť nasledujúce príkazy.

```
\begin{name}[text]
Toto je moja zaujímavá veta
\end{name}
```

Dost' bolo teórie. Nasledujúce príklady by, dúfajme, mali odstrániť aj posledné zvyšky pochybností a jasne ukázať, že prostredie `\newtheorem` je príliš zložitá cesta na pochopenie.

```
% deifnície pre dokument
% hlavička
\newtheorem{zakon}{Zákon}
\newtheorem{porota}[zakon]{Porota}
%v dokumente
\begin{zakon} \label{zakon:box}
Neskrývajte sa v~svedkovom boxe
\end{zakon}
\begin{porota}[Dvanásti]
Môžete to byť vy! Takže si dávajte
pozor a pozrite zákon
\ref{zakon:box}\end{porota}
\begin{zakon}Nie, Nie, Nie\end{zakon}
```

Zákon 1 *Neskrývajte sa v svedkovom boxe*

Porota 2 (Dvanásti) *Môžete to byť vy!
Takže si dávajte pozor a pozrite zákon 1*

Zákon 3 *Nie, Nie, Nie*

Teoréma „porota“ používa rovnaké číslovanie, ako teoréma „zákon“. Preto dostane poradové číslo, ktoré nasleduje za uvedenými teorémami „zákon“. Argument v hranatých zátvorkách sa používa na bližšie určenie názvu alebo niečoho podobného danej teoréme.

```
\newtheorem{mur}{Murphy}[section]
\begin{mur}
Ak existujú dva, alebo
viac spôsobov ako niečo
urobiť a jeden z týchto
spôsobov môže spôsobiť
katastrofu, potom si niekto
tento spôsob vyberie.\end{mur}
```

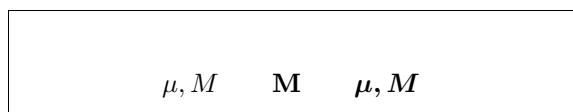
Murphy 3.8.1 *Ak existujú dva, alebo viac spôsobov ako niečo urobiť a jeden z týchto spôsobov môže spôsobiť katastrofu, potom si niekto tento spôsob vyberie.*

„Murphyho“ veta dostane číslo, ktoré sa viaže na číslo aktuálneho oddielu. Môžete použiť aj iné celky, ako napríklad kapitola (chapter) alebo pododdiel (subsection).

3.9 Tučné znaky

V \LaTeX u je pomerne zložitú dosiahnuť tučné znaky: je to tak asi úmyselne, pretože amatérski sadzači ich zvyknú nadmerne používať. Príkaz na zmenu fontov `\mathbf` síce dáva tučné písmo, ale to je roman (vzpriamené), kým matematické symboly sú obyčajne italic (naklonené). Existuje príkaz `\boldmath`, ale *funguje iba mimo matematického režimu*. Funguje aj pre znaky.

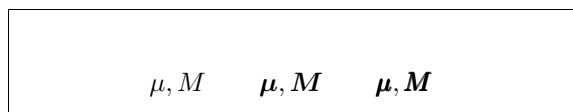
```
\begin{displaymath}
\mu, M \quad \mathbf{M} \quad \mu, M
\end{displaymath}
```



Všimnite si, že čiarka je tučná tiež, čo nemusí byť žiadané.

Balík `amsbsy`, (obsiahnutý v `amsmath`), toto veľmi zjednodušuje. Zahŕňa príkazy `\boldsymbol`, a „poor man’s bold“ `\pmb`, ktorého účinok je zvlášť významný u systémov, ktoré nemajú fonty, potrebné pre tučné symboly.

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



3.10 Zoznam matematických symbolov

V nasledujúcich tabuľkách nájdete zoznam všetkých symbolov bežne dostupných v *matematickom režime*.

Aby ste mohli použiť symboly uvedené v tabuľkách 3.12–3.16⁷, musíte to záhlavia dokumentu natiahnuť balík `amssymb` a musíte mať vo svojom systéme nainštalované matematické AMS fonty. V prípade, že uvedený balík ani fonty vo vašom systéme nainštalované nemáte, sú dostupné na adrese `CTAN:/tex-archive/macros/latex/packages/amslatex`

Tabuľka 3.1: Diakritika v matematickom režime

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

Tabuľka 3.2: Malé grécke písmená

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>		

Tabuľka 3.3: Veľké grécke písmená

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>		

⁷Tieto tabuľky boli odvodené zo súboru `symbols.tex`, vytvoreného Davidom Carlisleom a neskôr zmeneného a doplneného na návrh Jozefa Tkadleca

Tabuľka 3.4: Binárne relácie

Príslušné negácie môžete vytvoriť prídanim príkazu `\not` ako predpony pred nasledujúce symboly.

$<$	<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^a	<code>\sqsubset^a</code>	\in	<code>\in</code>	\bowtie	<code>\Join^a</code>
\sqsubseteq	<code>\sqsubseteq</code>	\vdash	<code>\vdash</code>	\bowtie	<code>\bowtie</code>
\sqsupset^a	<code>\sqsupset^a</code>	\ni , \owns	<code>\ni</code> , <code>\owns</code>	\propto	<code>\propto</code>
\sqsupseteq	<code>\sqsupseteq</code>	\dashv	<code>\dashv</code>	\models	<code>\models</code>
$ $	<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 or \ne	<code>\neq</code> or <code>\ne</code>

^aNa dosiahnutie tohto symbolu použite balík `latexsym`

Tabuľka 3.5: Binárne operátory

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

Tabuľka 3.6: Veľké operátory

\sum	<code>\sum</code>	\bigcup	<code>\bigcup</code>	\bigvee	<code>\bigvee</code>	\bigoplus	<code>\bigoplus</code>
\prod	<code>\prod</code>	\bigcap	<code>\bigcap</code>	\bigwedge	<code>\bigwedge</code>	\bigotimes	<code>\bigotimes</code>
\coprod	<code>\coprod</code>	\bigsqcup	<code>\bigsqcup</code>			\bigodot	<code>\bigodot</code>
\int	<code>\int</code>	\oint	<code>\oint</code>			\biguplus	<code>\biguplus</code>

Tabuľka 3.7: Šípky

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

^aNa dosiahnutie tohto symbolu použite balík `latexsym`

Tabuľka 3.8: Oddeľovače

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

Tabuľka 3.9: Veľké oddeľovače

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

Tabuľka 3.10: Rôzne znaky

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

^aNa dosiahnutie tohto symbolu použite balík `latexsym`

Tabuľka 3.11: Nematematické znaky

Tieto znaky sa môžu používať aj v textovom režime.

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

Tabuľka 3.12: AMS oddeľovače

\ulcorner	<code>\ulcorner</code>	\urcorner	<code>\urcorner</code>	\llcorner	<code>\llcorner</code>	\lrcorner	<code>\lrcorner</code>
-------------	------------------------	-------------	------------------------	-------------	------------------------	-------------	------------------------

Tabuľka 3.13: AMS grécke a hebrejské znaky

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

Tabuľka 3.14: AMS binárne relácie

\doteq	<code>\doteqdot</code> or <code>\Doteq</code>	\lessdot	<code>\lessdot</code>	\gtrdot	<code>\gtrdot</code>
\leqslant	<code>\leqslant</code>	\geqslant	<code>\geqslant</code>	\risingdotseq	<code>\risingdotseq</code>
\leqslantless	<code>\leqslantless</code>	\geqslantgtr	<code>\geqslantgtr</code>	\fallingdotseq	<code>\fallingdotseq</code>
\leqq	<code>\leqq</code>	\geqq	<code>\geqq</code>	\eqcirc	<code>\eqcirc</code>
\lll or \lllless	<code>\lll</code> or <code>\lllless</code>	\circeq	<code>\circeq</code>	\lesssim	<code>\lesssim</code>
\ggg or \gggtr	<code>\ggg</code> or <code>\gggtr</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>
\Subset	<code>\Subset</code>	\Supset	<code>\Supset</code>	\Vvdash	<code>\Vvdash</code>
\blacktriangleleft	<code>\blacktriangleleft</code>	\sqsubset	<code>\sqsubset</code>	\sqsupset	<code>\sqsupset</code>
\blacktriangleright	<code>\blacktriangleright</code>	\because	<code>\because</code>	\varpropto	<code>\varpropto</code>
\vartriangleright	<code>\vartriangleright</code>	\shortparallel	<code>\shortparallel</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>	\backepsilon	<code>\backepsilon</code>
\trianglelefteq	<code>\trianglelefteq</code>	\smallsmile	<code>\smallsmile</code>	\therefore	<code>\therefore</code>

Tabuľka 3.15: AMS šípky

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

Tabuľka 3.16: AMS negované binárne relácie a šípky

\nless	\ngtr	\varsubsetneqq
\lneq	\gneq	\varsupsetneqq
\nleq	\ngeq	\subsetneqq
\nleqslant	\ngeqslant	\supsetneqq
\lneqq	\gneqq	\mid
\lvertneqq	\gvertneqq	\parallel
\nleqq	\ngeqq	\shortmid
\lnsim	\gnsim	\shortparallel
\lnapprox	\gnapprox	\sim
\nprec	\nsucc	\cong
\npreceq	\nsucceq	\nvdash
\nprecneqq	\nsuccneqq	\nvDash
\nprecnsim	\succnsim	\nVdash
\nprecnapprox	\succnapprox	\nVDash
\subsetneq	\supsetneq	\triangleleft
\varsubsetneq	\varsupsetneq	\triangleright
\subsetneqq	\supsetneqq	\trianglelefteq
\subsetneqq	\supsetneqq	\trianglerighteq
\nleftarrow	\rightarrow	\leftrightarrow
\nLeftarrow	\Rightarrow	\Leftrightarrow

Tabuľka 3.17: AMS binárne operátory

\dotplus	\centerdot	\intercal
\ltimes	\rtimes	\divideontimes
\barwedge	\Cap or \doublecap	\boxminus
\veebar	\Cup or \doublecup	\boxdot
\boxplus	\smallsetminus	\circleddash
\boxtimes	\barwedge	\circledcirc
\leftthreetimes	\rightthreetimes	\circledast
\curlyvee	\curlywedge	

Tabuľka 3.18: AMS rôzne

\hbar	<code>\hbar</code>	\complement	<code>\complement</code>	\mathbb{k}	<code>\Bbbk</code>
\hslash	<code>\hslash</code>	■	<code>\blacksquare</code>	Ⓢ	<code>\circledS</code>
\vartriangle	<code>\vartriangle</code>	▲	<code>\blacktriangle</code>	□	<code>\square</code>
∇	<code>\triangledown</code>	▼	<code>\blacktriangledown</code>	⊃	<code>\Game</code>
\lozenge	<code>\lozenge</code>	◆	<code>\blacklozenge</code>	★	<code>\bigstar</code>
\sphericalangle	<code>\angle</code>	∠	<code>\measuredangle</code>	\	<code>\diagdown</code>
\diagup	<code>\diagup</code>	∠	<code>\sphericalangle</code>	⊣	<code>\Finv</code>
\nexists	<code>\nexists</code>	∖	<code>\backprime</code>	∪	<code>\mho</code>
\eth	<code>\eth</code>	∅	<code>\varnothing</code>		

Tabuľka 3.19: Matematické abecedy

Príklad	Príkaz	Potrebný balík
$ABCdef$	<code>\mathrm{ABCdef}</code>	
$ABCdef$	<code>\mathit{ABCdef}</code>	
$ABCdef$	<code>\mathnormal{ABCdef}</code>	
ABC	<code>\mathcal{ABC}</code>	
\mathcal{ABC}	<code>\mathcal{ABC}</code>	<code>mathrsfs</code>
ABC	<code>\mathcal{ABC}</code>	eucal s: <code>mathcal</code> alebo
	<code>\mathscr{ABC}</code>	eucal s: <code>mathscr</code>
\mathfrak{ABCdef}	<code>\mathfrak{ABCdef}</code>	<code>eufrak</code>
\mathbb{ABC}	<code>\mathbb{ABC}</code>	<code>amsfonts</code> alebo <code>amssymb</code>

Kapitola 4

Špeciality

Pri spájaní veľkého dokumentu vám \LaTeX pomôže riešiť niektoré špeciálne problémy akými sú vytváranie registra, memezment bibliografických údajov a iné veci. Podstatne úplnejší popis špecialít a rozšírení, prístupných v \LaTeX u môžete nájsť v *\LaTeX Manual* [1] a *The \LaTeX Companion* [3].

4.1 Vkladanie EPS grafiky

Pomocou prostredí `figure` a `table` poskytuje \LaTeX základné možnosti na prácu s plávajúcimi objektmi, napríklad s obrázkami alebo grafikou.

Pomocou \LaTeX u alebo \LaTeX ových doplnujúcich balíkov je možné vytvárať aj obrázky. Nanešťastie väčšina užívateľov považuje tento spôsob za náročný na pochopenie. Z tohoto dôvodu sa ďalej nebudeme vytváraním obrázkov v \LaTeX u v tomto manuále zaoberať.¹ Bližšie informácie o tvorbe obrázkov nájdete v *The \LaTeX Companion* [3] a *\LaTeX Manual* [1].

Oveľa jednoduchším spôsobom, ako dostať grafiku do dokumentu, je vytvoriť si daný obrázok špeciálnym grafickým programom,² a potom vložiť hotový obrázok do dokumentu. \LaTeX tu znovu ponúka širokú škálu balíkov, ktoré ponúkajú množstvo spôsobov, ako to urobiť. V tomto úvode sa budeme zaoberať len typom ENCAPSULATED POSTSCRIPT (EPS), nakoľko je pomerne jednoduchý a často používaný. Aby ste mohli používať obrázky vo formáte EPS, musíte mať k dispozícii POSTSCRIPTOVÚ tlačiareň.³

Prospešnú sadu príkazov na vkladanie grafiky poskytuje balík `graphicx`, ktorého autorom je D. P. Carlisle. Je to časť celej rodiny balíkov nazývanej „grafický“ uzol.⁴

¹V slovenskom preklade sa budeme, pozri oddiel 4.2 na strane 61. (pozn. prekl.)

²Ako napríklad XFig, CorelDraw!, Freehand, Gnuplot, ...

³Druhý spôsob, ako využiť POSTSCRIPT, je pomocou programu GHOSTSCRIPT, dostupného na adrese CTAN:/tex-archive/support/ghostscript

⁴CTAN:/tex-archive/macros/latex/packages/graphics

Ak predpokladáme, že pracujete na systéme, ktorý má k dispozícii POSTSCRIPTOVÚ tlačiareň a má nainštalovaný balík `graphicx`, tak môžete používať nasledujúci návod, kde sa krok za krokom dozviete, ako vložiť grafiku do dokumentu:

1. Uložte svoj obrázok v grafickom programe vo formáte EPS.⁵
2. Aktivujte balík `graphicx` v hlavičke zdrojového súboru príkazom

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

driver je názov vášho „dvi to postscript“ prekladača.⁶ Tento názov je balíkom požadovaný, pretože neexistuje štandard na včlenenie grafiky do T_EXu. Poznajúc *driver*, balík `graphicx` vloží správne metódu na vloženie informácie o obrázku do súboru `.dvi` tak, aby ju tlačiareň pochopila a mohla správne začleniť EPS súbor.

3. Na vloženie *súboru* do dokumentu použite príkaz

```
\includegraphics[klúč=value, ...]{file}
```

Za voliteľné parametre sa berie zoznam čiarkami oddelených *klúčov* a ich hodnôt. *Kľúče* sa môžu použiť na zmenu šírky, výšky a otáčanie vkladanej grafiky. Tabuľka 4.1 uvádza zoznam najdôležitejších kľúčov.

Tabuľka 4.1: Názvy kľúčov pre balík `graphicx`

<code>width</code>	zmení obrázok na danú šírku
<code>height</code>	zmení obrázok na danú výšku
<code>angle</code>	otočí obrázok v smere hodinových ručičiek
<code>scale</code>	zmení mierku obrázka

Nasledujúci príklad, dúfajme, urobí vo veciach jasno:

```
\begin{figure}
\begin{center}
```

⁵Ak váš software nedokáže exportovať do formátu EPS, môžete si skúsiť nainštalovať ovládač POSTSCRIPTovej tlačiarne (napr. nejaký Apple LaserWriter) a potom tlačiť s týmto ovládačom do súboru. Pri troche šťastia bude tento súbor vo formáte EPS. Poznamenajme, že EPS obrázok nesmie obsahovať viac ako jednu stranu. Niektoré ovládače tlačiarní môžu byť explicitne nakonfigurované na vytváranie formátu EPS.

⁶Najrozšírenejší program sa volá `dvips`.

```

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

```

Toto vloží obrázok, uložený v súbore `test.eps`. Obrázok je *najprv* otočený o 90 stupňov, a *potom* rozšírený na konečnú šírku rovnajúcu sa polovici šírky štandardného odstavca. Pomer výšky a šírky sa nezmenil, nakoľko sme nedefinovali žiadnu špeciálnu výšku. Parametre šírky a výšky môžu byť tiež zadané v absolútnych jednotkách. Pozrite sa na tabuľku 5.4 na strane 78. Ak sa o tejto téme chcete dozvedieť viac, určite si prečítajte [8] a [11].

4.2 Prostredie picture

Na tomto mieste doplníme stručný popis prostredia `picture`.⁷ Podrobnejší popis nájdete v knižke [12]. \LaTeX ovské prostredie `picture` umožňuje priamo v \LaTeX u vytvárať jednoduché obrázky, pozostávajúce z rovných čiar, šípiek, kružníc a oválov (resp. polkružníc, štvŕfkružníc). Okrem toho vyrába Bezie-rove krivky. Toto prostredie je vhodné aj na umiestňovanie popisov a to aj popisov ku grafike, vytvárajanej mimo \TeX u, napríklad k bitmapovým obrázkom. Takýmto spôsobom sa dosiahne jednota textu a popisov obrázkov. Nasledujúci obrázok 4.1 ilustruje uvedené možnosti. Bol vytvorený programom \TeX cad 3.2 (autori G. Horn a J. Winkelmann), ktorý je aj súčasťou distribúcie \CTEX u. Tento program je vhodný na prvé zoznámenie s prostredím `picture`, užívateľ môže neskôr písať príkazy samostatne. Pri použití príkazov `\emline` a `\bezier` sú potrebné štýly `emlines2.sty` a `bezier.sty`. Navyše nie všetky ovládače zobrazia výsledok príkazu `\emline`.

Syntax prostredia je približne nasledujúca:

```

\begin{picture}(šírka,výška)(x_r,y_r)
\put(x,y){objekt}
\end{picture}

```

Parametre *šírka* a *výška* udávajú rozmery vytvoreného „boxu“. Zadanie $(0,0)$ spôsobí, že vytvorený objekt má nulový rozmer, teda \TeX ostáva na mieste, kde bol pred zadaním prostredia `picture`. Nepovinné (x_r, y_r) sú súradnice „referenčného bodu“, teda bodu, v ktorom sa momentálne \TeX nachádza, v súradnicovom systéme prostredia `picture`.

```

Uprostred slo\begin{picture}(0,0)(0,0)
\unitlength=1mm
\put(2,2.5)
{\framebox(4,2.5)[tr]{\bullet}}
\end{picture}va môžem na chvíľu odísť.

```

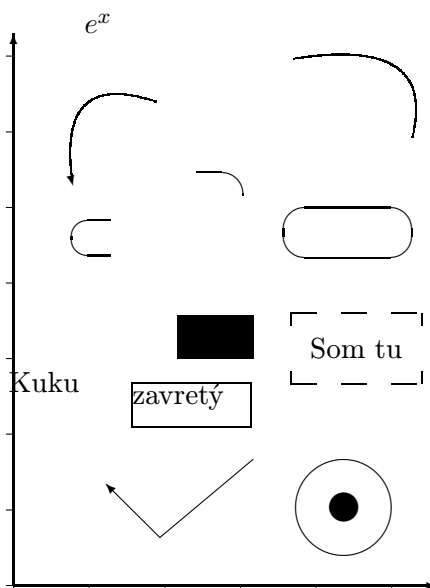
Uprostred slova môžem na chvíľu odísť.

⁷Doplnené pri slovenskom preklade (pozn. prekl.)

```

\special{em:linewidth 0.4pt} \unitlength 1mm \linethickness{0.4pt}
\begin{picture}(56.00,73.00)
\put(0,0){\vector(1,0){56}} \put(0,0){\vector(0,1){73}} % osi x,y
\multiput(0,0)(10,0){6}{\emline{0}{0}{1}{0}{-1}{1}} % znacky osi x
\multiput(0,0)(0,10){8}{\emline{0}{0}{1}{-1}{0}{1}} % znacky osi y
\put(8.67,28.33){\makebox(0,0)[rt]{Kuku}}
\put(15.67,21.00){\framebox(15.67,5.67)[lt]{Zavretý}}
\put(36.67,26.67){\dashbox{3.33}(17.33,9.33)[cc]{Som tu}}
\put(21.67,30.00){\rule{10.00\unitlength}{5.67\unitlength}}
\emline{31.67}{16.67}{1}{19.33}{6.33}{2}
\put(12.33,13.33){\vector(-1,1){0.2}}
\emline{19.33}{6.33}{3}{12.33}{13.33}{4} %\end
\put(43.67,10.33){\circle{13.33}} \put(43.67,10.33){\circle*{4.00}}
\put(12.83,46.00){\oval(10.33,4.67)[l]}
\put(24.17,51.50){\oval(12.33,6.33)[rt]}
\put(44.17,46.67){\oval(17.00,6.67)[]}
\bezier{132}(52.67,59.33)(56.00,72.67)(37.00,69.67)
\bezier{116}(18.67,64.00)(5.67,68.33)(7.67,53.67)
\put(7.67,53.67){\vector(1,-4){0.2}} \put(9.33,73.00){\mathex}
\end{picture}

```



Obr. 4.1: Príklad použitia prostredia picture

4.3 Zoznam použitej literatúry

Prostredím `thebibliography` môžete vytvárať zoznam použitej literatúry. Každý záznam sa začína príkazom

```
\bibitem{značka}
```

Značka sa potom používa na citovanie knihy v dokumente.

```
\cite{Značka}
```

Číslovanie záznamov je automatické. Maximálnu šírku (maximálny počet) týchto čísel nastavuje parameter za príkazom `\begin{thebibliography}`. V nižšie uvedenom príklade oznamuje `{99}` L^AT_EXu aby očakával, že žiadny bibliografický člen nebude širší ako číslo 99.

```
Partl~\cite{pa}
navrhol, že \ldots
```

```
\begin{thebibliography}{99}
\bibitem{pa} H.~Partl:
\emph{German \TeX},
TUGboat Vol.~9, No.~1 ('88)
\end{thebibliography}
```

Partl [1] navrhol, že ...

Literatúra

[1] H. Partl: *German T_EX*, TUGboat Vol. 9, No. 1 ('88)

Pri väčších projektoch sa vám môže hodiť program BibT_EX. Tento je súčasťou väčšiny T_EXovských distribúcií. Umožní vám udržiavať bibliografickú databázu a potom z nej vyberať odkazy, týkajúce sa vecí, ktoré citujete vo vašom článku. Vizuálna prezentácia literatúry vytvorenej BibT_EXom je založená na koncepcii štýlov, čo vám umožní vytvárať literatúru podľa požiadaviek širokej množiny rôznych zriadených vzorov.

Tabuľka 4.2: Príklady syntaxe indexového kľúča

Príklad vstupu	Výstup	Poznámka
<code>\index{hello}</code>	hello, 1	základná položka
<code>\index{hello!Peter}</code>	Peter, 3	podpoložka položky ‘hello’
<code>\index{Sam@\textsl{Sam}}</code>	<i>Sam</i> , 2	formátovaný vstup
<code>\index{Lin@\textbf{Lin}}</code>	Lin , 7	podobne ako vyššie
<code>\index{Jenny textbf}</code>	Jenny, 3	formát čísla strany
<code>\index{Joe textit}</code>	Joe, <i>5</i>	podobne ako vyššie

4.4 Tvorba registrov

Veľmi prospešnou stránkou mnohých kníh je ich register. \LaTeX om a podporným programom `makeindex`⁸ sa registre dajú vytvárať pomerne jednoducho. V tomto návode budú vysvetlené len základné príkazy na tvorbu registrov. Hlbšie znalosti získate z knižky *The \LaTeX Companion* [3].

Aby sa samotný register mohol vytvárať, v hlavičke musí byť natiahnutý balík `makeidx`:

```
\usepackage{makeidx}
```

a špeciálne príkazy na vytváranie registra musia byť aktivované umiestnením príkazu

```
\makeindex
```

do hlavičky zdrojového súboru.

Obsah registra sa určuje príkazmi

```
\index{klíč}
```

kde *klíč* je záznam v registri. Príkazy na tvorbu registru umiestňujete na miesto v texte, na ktoré chcete, aby index odkazoval. Tabuľka 4.2 popisuje syntax parametra *klíč* s niekoľkými príkladmi.

Keď \LaTeX prekladá zdrojový súbor, každý príkaz `\index` zapisuje príslušný indexový záznam spolu s číslom strany do špeciálneho súboru. Tento súbor má rovnaký názov ako \LaTeX ový zdrojový dokument, avšak líši sa koncovkou (`.idx`). Tento `.idx` súbor potom môže byť spracovaný programom

⁸V systémoch, ktoré nepodporujú názvy dlhšie ako 8 znakov sa môže nazývať `makeidx`. Program `csindex`, vytvorený pánom Z. Wagnerom, je česká/slovenská implementácia programu `makeindex` (pozn. prekl.)

makeindex:

```
makeindex filename
```

Tento vytvorí usporiadaný register s rovnakým názvom, avšak v tomto prípade s príponou `.ind`. V prípade, ak bude \LaTeX tento zdrojový súbor znovu prekladať, bude tento zoradený register umiestnený v dokumente na mieste, kde sa nachádza príkaz

```
\printindex
```

Balík `showidx`, ktorý je distribuovaný spolu s \LaTeX 2 ϵ , vytlačí všetky indexované údaje na ľavom okraji textu. Toto je celkom prospešné pri kontrolnom čítaní dokumentu a overovaní registru.

4.5 Exotické hlavičky

Balík `fancyhdr` (od spojenia Fancy Headers),⁹ napísaný Pietom van Oostrum, poskytuje niekoľko jednoduchých príkazov, ktoré vám umožnia prispôbiť si riadky v hlavičke alebo v päte vášho dokumentu. Ak sa pozriete na vrch strany, ktorú práve čítate, môžete vidieť možné použitie tohto balíka.

Skryté a neočakávané ťažkosti sa môžu vyskytnúť keď sa snažíme hlavičky alebo päty dostať niečo ako priebežné oddiely a kapitoly. \LaTeX to vykoná dvojstupňovým prístupom. V definícii hlavičky a päty použijete príkazy `\rightmark` a `\leftmark` na reprezentáciu názvu aktuálnej kapitoly, resp. oddielu. Hodnoty týchto dvoch príkazov budú prepísané pri každom vykonaní príkazu `\chapter` alebo `\section`.

Kvôli konečnej pružnosti príkaz `\chapter` s kamarátmi nepredefinujú samotné `\rightmark` a `\leftmark`. Volajú ešte ďalšie príkazy `\chaptermark`, `\sectionmark` alebo `\subsectionmark`, ktoré sú zodpovedné za predefinovanie obsahu `\rightmark` a `\leftmark`.

Takže ak chcete zmeniť vzhľad názvu kapitoly v hlavičkovom riadku, musíte jednoducho „obnoviť“ príkaz `\chaptermark`.

Obrázok 4.2 ukazuje možné nastavenia pre balík `fancyhdr`, ktoré vytvoria hlavičky, vyzerajúce približne tak ako hlavičky v tejto brožúrke. V každom prípade vám doporučujem stiahnuť si dokumentáciu k balíku na adrese, uvedenej v poznámke.

4.6 Balík verbatim

V našej knižke ste sa už skôr zoznámili s *prostredím verbatim*. V tomto oddiele sa naučíte niečo o *balíku verbatim*. Balík `verbatim` je v princípe re-

⁹Dostupný na CTAN: [/tex-archive/macros/latex/contrib/supported/fancyhdr](http://tex-archive/macros/latex/contrib/supported/fancyhdr).

```

\documentclass{book}
\usepackage{fancyhdr}
\pagestyle{fancy}
% týmto sa uistíme, že hlavičky pre chapter a section
% budú písané malými písmenami.
\renewcommand{\chaptermark}[1]{\markboth{#1}{}}
\renewcommand{\sectionmark}[1]{\markright{\thesection\ #1}}
\fancyhf{} % ruší aktuálne nastavenia pre hlavičku a päť
\fancyhead[LE,R0]{\bfseries\thepage}
\fancyhead[L0]{\bfseries\rightmark}
\fancyhead[RE]{\bfseries\leftmark}
\renewcommand{\headrulewidth}{0.5pt}
\renewcommand{\footrulewidth}{0pt}
\addtolength{\headheight}{0.5pt} % vytvára medzeru pre čiaru
\fancypagestyle{plain}{%
  \fancyhead{} % zbaví sa hlavičky na stránkach plain
  \renewcommand{\headrulewidth}{0pt} % aj čiary
}

```

Obr. 4.2: Príklad nastavenia fancyhdr.

implementácia prostredia `verbatim`, ktorá obchádza niektoré ohraničenia pôvodného prostredia `verbatim`. Toto samo osebe nie je nič mimoriadne, ale s touto implementáciou balíka `verbatim` boli pridané aj nové funkcie, čo bolo dôvodom na to, že tu tento balík spomínam. Balík `verbatim` sprístupňuje príkaz

```
\verbatiminput{filename}
```

ktorý vám umožní začleniť surový ASCII text do vášho dokumentu ako keby bol vo vnútri prostredia `verbatim`.

Keďže balík `verbatim` je časťou uzla ‘tools’, mali by ste ho nájsť nainštalovaný na väčšine systémov. Ak sa o tomto balíku chcete dozvedieť viac, určite si prečítajte [9].

4.7 Ochrana krehkých príkazov

Text, zadany ako argument príkazov typu `\caption` alebo `\section` sa môže ukazovať v dokumente viackrát (napr. v obsahu rovnako ako v samotnom texte). Niektoré príkazy skrachujú, ak sú použité ako argument príkazov typu `\section`. Tieto príkazy sa nazývajú krehké príkazy. Krehký príkaz je napríklad `\footnote` alebo `\phantom`. Na to, aby tieto príkazy fungovali,

potrebujú ochranu (a my všetci nie?). Môžete ich ochrániť pridaním pred nich príkazu `\protect`.

`\protect` sa vzťahuje len na príkaz nasledujúci bezprostredne za ním, nevzťahuje sa sokonca ani na jeho argumenty. Nadbytočný príkaz `\protect` vo väčšine prípadov neuškodí.

```
\section{Som ohľaduplný  
      \protect\footnote{a chránim svoje poznámky pod čiarou}}
```


Kapitola 5

Prispôsobovanie L^AT_EXu

Vzhľad dokumentov, vytvorených na základe príkazov, ktoré ste sa doteraz naučili bude prijateľný pre široký kruh užívateľov. Hoci nemusia vyzerat' fantasticky, budú vyhovovať ustanoveným pravidlám dobrej typografie, čo ich urobí ľahko čitateľnými a na pohľad príjemnými.

Napriek tomu sú situácie, keď L^AT_EX neposkytuje príkazy alebo prostredia, odpovedajúce vašim potrebám, prípadne výstup, získaný na základe existujúcich príkazov, neuspokojuje vaše požiadavky.

V tejto kapitole sa pokúsím trochu naznačiť ako naučiť L^AT_EX nové triky a ako ho donútiť vytvoriť výstup, ktorý vyzerá ináč, ako to poskytuje štandard.

5.1 Nové príkazy, prostredia a balíky

Mohli ste si všimnúť, že všetky príkazy, ktoré som uviedol v tejto knihe, sú sádzané v rámečkoch a že sú vytlačené v registri na konci knižky. Namiesto priameho použitia príkazov L^AT_EXu, potrebných na dosiahnutie týchto úloh, som vytvoril balík, v ktorom som za týmto účelom definoval nové príkazy a prostredia. Teraz jednoducho napíšem:

```
\begin{command}  
\ci{dum}  
\end{command}
```



\dum

V tomto príklade používam oboje: nové prostredie s názvom `command`, ktoré je zodpovedné za nakreslenie rámčeka okolo príkazu aj nový príkaz `\ci`, ktorý vytlačí názov príkazu a zároveň vytvorí odpovedajúcu položku registra. Môžete sa o tom presvedčiť, ak sa pozriete do registra na konci knihy a nájdete položku príkazu `\dum`, ukazujúcu na každú stranu, kde som sa o príkaze `\dum` zmienil.

Ak by som sa hocikedy rozhodol, že už viac nemám rád príkazy, sádzané v rámečkoch, môžem jednoducho zmeniť definíciu prostredia `command` a vytvoriť jeho nový vzhľad. Je to oveľa ľahšie ako prechádzať celým dokumentom a hľadať všetky miesta, kde som použil niektoré základné príkazy L^AT_EXu na vykreslenie rámečku okolo nejakého slova.

5.1.1 Nové príkazy

Na pridanie vašich nových príkazov použite príkaz

```
\newcommand{meno}[num]{definícia}
```

Príkaz potrebuje v podstate dva argumenty. *meno* – názov príkazu, ktorý chcete vytvoriť a *definíciu* príkazu. Argument *num* v hranatých zátvorkách je nepovinný a určuje počet parametrov, ktoré používa nový príkaz (maximálne 9). Ak chýba, štandardná hodnota je 0, t. j. nie je dovolený žiaden argument.

Nasledujúce dva príklady by vám mali pomôcť pochopiť hlavnú myšlienku. Prvý príklad definuje príkaz nazývaný `\npsu` (toto je skratka pre „Nie príliš stručný úvod do systému L^AT_EX 2_ε“). Takýto príkaz môže byť veľmi praktický, keď musíte písať názov tejto knihy znovu a znovu.

```
\newcommand{\npsu}
  {Nie príliš stručný úvod
   do systému \LaTeXe}
% v dokumente:
Toto je \uv{\npsu} \ldots{} \uv{\npsu}
```

Toto je „Nie príliš stručný úvod do systému L^AT_EX 2_ε“ ... „Nie príliš stručný úvod do systému L^AT_EX 2_ε“

Nasledujúci príklad ukazuje ako sa definuje nový príkaz s parametrom. Značka #1 bude nahradená parametrom, ktorý zadáte. Ak chcete použiť viac ako jeden parameter, použijete #2, a tak ďalej.

```
\newcommand{\txsit}[1]
  {Toto je \emph{#1} stručný
   úvod do systému \LaTeXe}
% v dokumente:
\begin{itemize}
\item \txsit{nie príliš}
\item \txsit{veľmi}
\end{itemize}
```

- Toto je *nie príliš* stručný úvod do systému L^AT_EX 2_ε
- Toto je *veľmi* stručný úvod do systému L^AT_EX 2_ε

L^AT_EX vám nedovolí vytvoriť nový príkaz, ktorý predefinuje už existujúci. Ale v prípade, ak chcete urobiť práve toto, máte k dispozícii zvláštny príkaz `\renewcommand`. Používa rovnakú syntax ako príkaz `\newcommand`.

Za určitých podmienok môžete tiež chcieť použiť príkaz `\providecommand`. Funguje podobne ako `\newcommand`, ale ak príkaz s týmto názvom je už definovaný, L^AT_EX ho bude potichu ignorovať.

Pár slov by bolo dobré povedať o medzerách, nasledujúcich za príkazmi \LaTeX u. Kvôli podrobnejšej informácii pozri stranu 6.

5.1.2 Nové prostredia

Podobný príkaz ako \newcommand existuje aj na vytváranie vašich vlastných prostredí. Počas písania tohoto úvodu do \LaTeX u som si vytvoril špeciálne prostredia pre štruktúry, ktoré som opakovane používal: „príklady“, „úryvky zo zápisov“ a „rámčeky definícií príkazov“. Príkaz \newenvironment má nasledujúcu štruktúru:

```
 $\text{\newenvironment}{meno}[num]{pred}{potom}$ 
```

Ako pri príkaze \newcommand , aj príkaz \newenvironment môžete použiť s parametrom alebo bez neho. Materiál, bližšie určený v parametri *pred*, je spracovaný predtým, ako text v danom prostredí. Materiál, bližšie určený v parametri *potom*, je spracovaný až po vykonaní príkazu $\text{\end{meno}}$.

Nižšie uvedený príkaz ukazuje používanie príkazu \newenvironment .

```
 $\text{\newenvironment}{king}
{\rule{1ex}{1ex}%
 \hspace{\stretch{1}}}
{\hspace{\stretch{1}}%
 \rule{1ex}{1ex}}

% použite toto v časti textu
\begin{king}
Moje skromné prípady \ldots
\end{king}$ 
```

■ Moje skromné prípady ... ■

Argument *num* sa používa rovnakým spôsobom, ako to bolo v prípade príkazu \newcommand . \LaTeX spoľahlivo zabezpečuje, aby ste nenadefinovali prostredie, ktoré už existuje. Ak budete chcieť niekedy zmeniť existujúci príkaz, môžete použiť príkaz \renewenvironment . Používa sa rovnakým spôsobom ako príkaz \newenvironment .

Príkazy, použité v tomto príklade budú vysvetlené neskôr: Na príkaz \rule sa pozri na stranu 83, príkaz \stretch nájdeš na strane 77. Podrobnejšia informácia o \hspace sa dá nájsť na strane 77.

5.1.3 Váš vlastný balík

Ak definujete veľké množstvo prostredí a príkazov, bude preambula vášho dokumentu príliš dlhý. V tomto prípade je dobrým nápadom vytvorenie

L^AT_EXovského balíka, obsahujúceho všetky definície vašich príkazov a prostredí. Na sprístupnenie balíka vo vašom dokumente môžete použiť príkaz `\usepackage`.

```
% Demo Package by Tobias Oetiker
\ProvidesPackage{demopack}
\newcommand{\npsu}{Nie príliš stručný úvod do systému \LaTeXe}
\newcommand{\txsit}[1]{Toto je \emph{#1} stručný úvod
                      do systému \LaTeXe}
\newenvironment{king}{\rule{1ex}{1ex}%
                     \hspace{\stretch{1}}}{\hspace{\stretch{1}}\rule{1ex}{1ex}}
```

Obr. 5.1: Príklad balíka.

Napísanie balíka spočíva v podstate v skopírovaní obsahu vašej preamble do samostatného súboru s názvom s koncovkou `.sty`. Existuje jeden zvláštny príkaz,

`\ProvidesPackage{package name}`

používaný na úplnom začiatku súboru vášho balíka. `\ProvidesPackage` oznámi L^AT_EXu názov balíka a umožní mu vydať výrazné chybové hlásenie ak sa pokúsíte balík inicializovať druhý raz. Obrázok 5.1 ukazuje malý príklad balíka, obsahujúceho príkazy, definované vo vyššie uvedených príkladoch.

5.2 Fonty a veľkosti

5.2.1 Príkazy na zmenu písma

L^AT_EX volí vhodný font a jeho veľkosť na základe logickej štruktúry dokumentu (oddiely, poznámky pod čiarou, ...). V niektorých prípadoch môže niekto chcieť zmeniť fonty a ich veľkosti ručne. Toto sa dá urobiť pomocou príkazov, ktoré sú uvedené v tabuľkách 5.1 a 5.2. Skutočná veľkosť každého fontu je výsledok dizajnu a závisí od triedy dokumentu a jej volieb. Tabuľka 5.2 ukazuje absolútne bodové veľkosti pre tie príkazy, ktoré sú súčasťou štandardných tried dokumentu.

Tabuľka 5.1: Fonty

<code>\textrm{...}</code>	roman (antikva)	<code>\texttt{...}</code>	typewriter
<code>\textsf{...}</code>	sans serif (bezpätkové)		
<code>\textbf{...}</code>	bold face (tučné)	<code>\textmd{...}</code>	medium
<code>\textit{...}</code>	<i>italic</i> (kurzíva)	<code>\textup{...}</code>	upright
<code>\textsl{...}</code>	<i>slanted</i> (naklonené)	<code>\textsc{...}</code>	SMALL CAPS
<code>\emph{...}</code>	<i>emphasised</i>	<code>\textnormal{...}</code>	bežné

```
{\small The small and
\textbf{bold} Romans ruled}
{\Large all of great big
\textit{Italy}.}
```

```
{\small Malí,
\textbf{tuční} Rimania vládli}
{\Large celej veľkej
\textit{Itálie}.}%
\footnote{roman\,--\, antikva,
italic\,--\,kurzíva, bold\,--\,tučný
--- v~angličtine je to pekná
typografická slovná hračka (pozn. prekl.)}
```

The small and **bold** Romans ruled all of great big *Italy*.

Malí, **tuční** Rimania vládli celej veľkej *Itálie*.^a

^aroman – antikva, italic – kurzíva, bold – tučný — v angličtine je to pekná typografická slovná hračka (pozn. prekl.)

Jedna dôležitá vlastnosť systému $\text{\LaTeX} 2_{\epsilon}$ je, že vlastnosti fontov sú nezávislé. To znamená, že môžete zmeniť počiatočnú veľkosť, alebo dokonca typ fonu, a pritom zachovať skôr zadaný príkaz na tučné alebo naklonené písmo.¹

V *matematickom režime* môžete používať na dočasné opustenie *matematického režimu* a napísanie normálneho textu príkazy na zmenu aktuálneho typu písma. Ak chcete v matematickom režime použiť nejaký iný typ písma, existuje ďalšia sada príkazov. Tieto fonty sú podrobne uvedené v tabuľke 5.3.

V súvislosti s príkazmi na veľkosť a tvar písma majú významné postavenie zložené zátvorky. Používajú sa na vytváranie *skupín*. Skupiny obmedzujú pôsobenie väčšiny \LaTeX ových príkazov.

```
Má rád {\LARGE veľké a
\small malé} písmená.
```

Má rád veľké a malé písmená.

¹Aj keď je toto zrejme pre niekoho, kto sa učí $\text{\LaTeX} 2_{\epsilon}$, nie je to také zrejme pre niekoho, kto používal $\text{\LaTeX} 2.09$.

Tabuľka 5.2: Veľkosti fontov

<code>\tiny</code>	drobný font	<code>\Large</code>	väčší font
<code>\scriptsize</code>	veľmi malý font	<code>\LARGE</code>	veľmi veľký font
<code>\footnotesize</code>	dosť malý font	<code>\huge</code>	obrovský font
<code>\small</code>	malý font	<code>\Huge</code>	najväčší font
<code>\normalsize</code>	normálny font		
<code>\large</code>	veľký font		

Tabuľka 5.3: Matematické fonty

Príkaz	Príklad	Výstup
<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 R}\$</code>	$G \times R$
<code>\mathtt{...}</code>	<code>\$\$\mathtt{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>	$ffi \neq ffi$

Príkaz na zmenu veľkosti fontov môže zmeniť aj riadkovanie, avšak iba ak odstavec končí za účinku príkazu na veľkosť fontu. Zatváracia zložená zátvorka `}` preto nesmie prísť príliš skoro. Všimnite si polohu príkazu `\par` v nasledujúcich dvoch príkladoch.

```
{\Large Nečítajte to! To nie
je pravda. Môžete mi veriť!}\par}
```

Nečítajte to! To nie je pravda.
Môžete mi veriť!

```
{\Large Toto nie je pravda,
ale nezabúdajte: som klamár.}\par}
```

Toto nie je pravda, ale nezabú-
dajte: som klamár.

Ak chcete uplatniť príkaz na zmenu veľkosti pre celý odstavec textu alebo dokonca na viac textu, môže byť užitočné použiť na zmenu veľkosti syntax typu prostredí.


```
\begin{Large}
To nie je pravda. Ale potom
znova, čo sú dnešné dni
\ldots
\end{Large}
```

To nie je pravda. Ale potom
znova, čo sú dnešné dni ...

Toto vás ochráni pred počítaním množstva zložených zátvoriek.

5.2.2 Pozor, nebezpečenstvo!

Ako bolo poznamenané na začiatku tejto kapitoly, je nebezpečné preplniť váš dokument explicitnými príkazmi, pretože tieto potláčajú základnú myšlienku L^AT_EX, ktorou je oddelenie logického a vizuálneho značkovania vášho dokumentu. To znamená, že ak chcete použiť rovnaký príkaz na zmenu písma na viacerých miestach kvôli tomu, aby ste zvýraznili zvláštny druh informácií, mali by ste použiť príkaz `\newcommand` na definovanie „príkazu logického vyčlenenia“, ktorý zmení font.

```
\newcommand{\oops}[1]{\textbf{#1}}
\oops{Nevstupujte} do tejto
miestnosti, je obsadená \oops{strojom}
neznámeho pôvodu a účelu.
```

Nevstupujte do tejto miestnosti, je obsadená **strojom** neznámeho pôvodu a účelu.

Tento prístup má tú výhodu, že vám neskôr umožní rozhodnúť sa, či nechcete použiť nejakú inú vizuálnu reprezentáciu nebezpečenstva ako je použitie `\textbf` bez toho, aby ste sa brodili svojím dokumentom, vyhľadávali všetky výskyty príkazu `\textbf` a potom pre každý z nich určovali, či bol použitý na vyznačenie nebezpečenstva alebo na nejaký iný účel.

5.2.3 Rada

Na záver tejto cesty do krajiny fontov a ich veľkostí vám ponúkam malú radu:

Pamätajte! Čím **VI** AC fontov použijete vo vašom dokumente, tým *krajším* a ČITATEĽNEJŠÍM *sa stane*.²

²Pozor! Toto je trochu satira. Dúfam, že ste si to všimli!

5.3 Medzery

5.3.1 Riadkovanie

Ak chcete väčšie medzery medzi riadkami svojho dokumentu, môžete to urobiť pomocou príkazu

```
\linespread{faktor}
```

umiestneného v hlavičke dokumentu. Príkaz `\linespread{1.3}` zabezpečí riadkovanie $1\frac{1}{2}$, pomocou príkazu `\linespread{1.6}` dosiahneme riadkovanie 2. Obvykle sa medzery medzi riadkami nezväčšujú, preto je základný zväčšovací koeficient rovný 1.

5.3.2 Formátovanie odstavcov

V L^AT_EXu existujú dva príkazy ovplyvňujúce úpravu odstavcov. Pomocou príkazov

```
\setlength{\parindent}{0pt}
\setlength{\parskip}{1ex plus 0.5ex minus 0.2ex}
```

umiestnených v hlavičke³ zdrojového súboru, môžete meniť nastavenie odstavcov. Tieto dva príkazy zväčšujú miesto medzi odstavcami, pričom rušia odsadzovanie prvých riadkov odstavcov. V kontinentálnej Európe sa odstavce často oddeľujú voľným miestom a neodsadzujú sa ich prvé riadky. Ale pozor, toto má vplyv aj na obsah. Aj riadky obsahu budú oddeľované väčším priestorom. Aby ste sa tomu vyhli, môžete umiestniť tieto dva príkazy na nejaké miesto za príkazom `\tableofcontents` alebo sa môžete rozhodnúť, že ich nepoužijete vôbec, pretože ste zistili, že väčšina profesionálnych kníh používa odsadzovanie odstavcov bez medzery medzi nimi.

Ak chcete odsadiť odstavec, ktorý nie je odsadený, môžete použiť na začiatku odstavca príkaz⁴

```
\indent
```

Samozrejme, že to môže fungovať len v prípade, ak nie je `\parindent` nastavený na nulu.

³Medzi príkazy `\documentclass` a `\begin{document}`

⁴Ak chcete odsadiť prvý odstavec v každej kapitole, použite balík `indentfirst` z uzla `tools`.

Ak chcete, aby daný odstavec nebol odsadený, môžete na začiatku odstavca použiť príkaz

```
\noindent
```

Môže sa vám to zísť, ak začínate dokument samotným textom a nie príkazom na kapitolu, resp. oddiel.

5.3.3 Vodorovné medzery

L^AT_EX určuje medzery medzi slovami a vetami automaticky. Ak chcete zväčšiť vodorovnú (horizontálnu) medzeru, musíte použiť príkaz:

```
\hspace{dĺžka}
```

Ak sa má takáto medzera zachovať aj v prípade, že prípadne na koniec alebo začiatok riadku, použijete príkaz `\hspace*` namiesto `\hspace`. V najjednoduchších prípadoch je *dĺžka* len číslo a jednotka. Najdôležitejšie jednotky sú uvedené v tabuľke 5.4.

Táto`\hspace{1.5cm}`medzera
má 1.5 cm.

Táto	medzera má 1.5 cm.
------	--------------------

Príkaz

```
\stretch{n}
```

vytvára špeciálnu mäkkú (gumenú) medzeru. Rozpína sa, až kým nie je vyplnený celý zvyšný priestor riadku. Ak sa v jednom riadku nachádzajú dva príkazy `\hspace{\stretch{n}}`, narastajú primerane svojmu rozšírovaciemu koeficientu.

`x\hspace{\stretch{1}}`
`x\hspace{\stretch{3}}`x

x	x	x
---	---	---

5.3.4 Zvislé medzery

Medzera medzi odsekmi, oddielmi, pooddieľmi, ... je určovaná automaticky L^AT_EXom. V prípade, ak je to potrebné, môžete zvislú (vertikálnu) medzeru zväčšiť pomocou príkazu:

```
\vspace{length}
```

Tento príkaz sa normálne oddeľuje voľným riadkom pred a za príkazom.

Tabuľka 5.4: Jednotky T_EXu

mm	milimeter $\approx 1/25$ inch	⊐
cm	centimeter = 10 mm	┌───┐
in	inch = 25.4 mm	┌──────────┐
pt	point $\approx 1/72$ inch $\approx \frac{1}{3}$ mm	⊐
em	približne výška daného fontu	┌──┐
ex	približne výška x v danom fonte	┌──┐

V prípade, ak sa má medzera udržiavať aj na začiatku (konci) strany a medzi dvoma stránkami, použite namiesto `\vspace` hviezdíčkovú verziu príkazu `\vspace*`.

Príkaz `\stretch` spolu s príkazom `\pagebreak` sa môže použiť na napísanie textu na posledný riadok strany, prípadne na zvislé vycentrovanie strany.

Nejaký text `\ldots`

`\vspace{\stretch{1}}`

Toto bude napísané na poslednom riadku strany.`\pagebreak`

V prípade, ak chceme zväčšiť medzery medzi riadkami *toho istého* odstavca, alebo vo vnútri tabuľky, môžeme namiesto príkazu `\\` použiť príkaz:

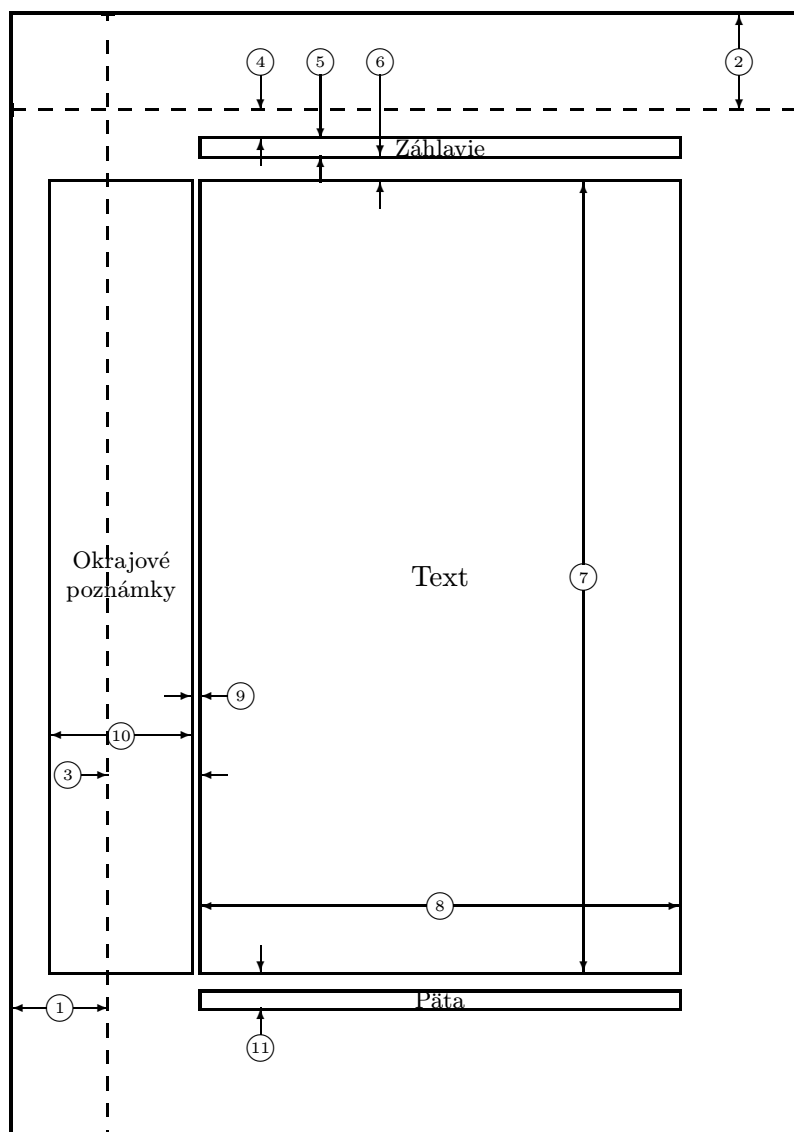
`\\[length]`

5.4 Rozvrhnutie stránky

L^AT_EX 2_ε vám umožňuje zadať veľkosť papiera v príkaze `\documentclass`. L^AT_EX potom automaticky zvolí správne rozmery textu. V niektorých prípadoch vám však preddefinované hodnoty nemusia vyhovovať. Prirodzene, môžete ich zmeniť. Obrázok 5.2 ukazuje všetky parametre, ktoré sa dajú meniť. Obrázok bol vytvorený pomocou balíka `layout` z uzla `tools`.⁵

POČKAJTE! ... skôr než sa pustíte do šialených aktivít typu: „Rozšírime trošku túto úzku stranu!“, porozmýšľajte pár sekúnd. Rovnako ako v prípade väčšiny vecí v L^AT_EXu, nájdete sa dobré dôvody aj nato, aby bol vzhľad stránky taký, aký je.

⁵CTAN:/tex-archive/macros/latex/packages/tools



1	jeden palec + <code>\hoffset</code>	2	jeden palec + <code>\voffset</code>
3	<code>\evensidemargin = 70pt</code>	4	<code>\topmargin = 22pt</code>
5	<code>\headheight = 13pt</code>	6	<code>\headsep = 19pt</code>
7	<code>\textheight = 595pt</code>	8	<code>\textwidth = 360pt</code>
9	<code>\marginparsep = 7pt</code>	10	<code>\marginparwidth = 106pt</code>
11	<code>\footskip = 27pt</code>		<code>\marginparpush = 5pt</code> (nezobrazené)
	<code>\hoffset = 0pt</code>		<code>\voffset = 0pt</code>
	<code>\paperwidth = 597pt</code>		<code>\paperheight = 845pt</code>

Obr. 5.2: Parametre ovplyvňujúce rozvrhnutie stránky

Samozrejme, v porovnaní so stránkou MS Wordu vyzerá hrozne úzka. Ale pozrite sa na vašu obľúbenú knihu⁶ a spočítajte počet písmen na štandardnom textovom riadku. Zistíte, že tam nie je viac písmen ako okolo 66. Teraz urobte to isté so svojou L^AT_EXovskou stranou. Zistíte, že aj tam sa nachádza približne 66 znakov na riadok. Skúsenosti ukazujú, že ak sa na riadku vyskytuje viac písmen, čítanie sa stáva obtiažnejším. Je to preto, lebo očiam je náročné presúvať sa z konca riadku na začiatok ďalšieho. Toto je tiež dôvod nato, aby sa noviny tlačili do viacerých stĺpcov.

Teda ak zväčšíte šírku vášho textu, uvedomte si, že komplikujete život čitateľom vašej práce. Ale dosť bolo výstrah, sľúbil som, že vám poviem, ako sa to robí . . .

L^AT_EX poskytuje dva príkazy na zmenu týchto parametrov. Väčšinou sa používajú v hlavičke dokumentu.

Prvým príkazom sa nastavuje určitá hodnota ktoréhokoľvek z daných parametrov:

```
\setlength{parameter}{length}
```

Druhý príkaz pridáva určitú hodnotu ku ktorémukoľvek z daných parametrov:

```
\addtolength{parameter}{length}
```

Tento druhý príkaz je v skutočnosti prospešnejší než príkaz `\setlength`, pretože môžete pracovať podobne, ako pri existujúcich nastaveniach. Aby ste k šírke stránky pridali jeden centimeter, do hlavičky treba pridať nasledujúce príkazy:

```
\addtolength{\hoffset}{-0.5cm}
\addtolength{\textwidth}{1cm}
```

V tejto súvislosti vás možno zaujme balík `calc`, umožňujúci použitie aritmetických operácií v argumente príkazu `\setlength` a na iných miestach, kde môžete zadávať numerické hodnoty ako argumenty funkcií.

5.5 Viac zábavy s dĺžkami

Vždy keď je to možné, vyhýbam sa používaniu absolútnych dĺžok v L^AT_EXovských dokumentoch. Radšej sa snažím založiť veci na dĺžke, výške alebo inom prvku strany. Pre šírku obrázku to môže byť `\textwidth` – takto zaplní stranu.

⁶Mám na mysli skutočnú tlačenú knihu vyrobenú vydavateľom s dobrou povestou.

Nasledujúce 3 príkazy umožňujú určiť šírku, výšku a hĺbku textového reťazca.

```
\settoheight{command}{text}
\settodepth{command}{text}
\settowidth{command}{text}
```

Nasledujúci príklad ilustruje možné použitie týchto príkazov.

```
\flushleft
\newenvironment{vardesc}[1]{%
  \settowidth{\parindent}{#1:\ }
  \makebox[Opt][r]{#1:\ }}{ }

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

\begin{vardesc}{Kde}$a$,
$b$ -- sú odvesny pravouhlého
trojuholníka.

$c$ -- je prepona trojuholníka
a cíti sa osamelo.

$d$ -- sa tu nakoniec vôbec nevyskytuje. Nie je to popletené?
\end{vardesc}
```

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

Kde: a , b – sú odvesny pravouhlého trojuholníka.

c – je prepona trojuholníka a cíti sa osamelo.

d – sa tu nakoniec vôbec nevyskytuje. Nie je to popletené?

5.6 Škatulky, krabičky, schránky, skrinky, boxy

\LaTeX vytvára svoje strany rozmiestňovaním boxov (škatuliek, krabičiek, schránok, skriniek). Po prvé, každé písmeno je malá krabička, ktorá sa ďalej nalepuje na iné písmená, vytvárajúc slová. Tieto sa zasa prilepujú k ďalším slovám, ale špeciálnym glejom⁷, ktorý je pružný. Preto môže byť rad slov stlačený alebo rozťahnutý tak, aby mohol prene vyplniť riadok na strane.

Pripúšťam, že toto je veľmi zjednodušená verzia toho, čo sa skutočne stane, ale myšlienka je taká, že \TeX narába s glejom (lepidlom) a so škatulkami (boxmi). Ale nielen písmená môžu tvoriť box. Do krabičky môžete položiť skutočne všetko, vrátane iných boxov. \LaTeX zaobchádza s každým boxom, akoby to bolo jedno písmeno.

V predchádzajúcich kapitolách ste sa už niektorými boxmi stretli, hoci som vám to nepovedal. Prostredie `tabular` a príkaz `\includegraphics`, napríklad, oboje vytvárajú box. To znamená, že ľahko môžete unietniť dve

⁷Anglicky glue (pozn. prekl.)

tabulky alebo dva obrázky vedľa seba. Len si musíte byť istí, že ich spoločná šírka nie je väčšia ako nastavená šírka strany.

Rovnako môžete do boxu zabaliť vami zvolený odstavec a to buď príkazom

```
\parbox[pos]{width}{text}
```

alebo pomocou prostredia

```
\begin{minipage}[pozícia]{šírka} text \end{minipage}
```

Parameter *pozícia* môže byť jedno z písmen *c*, *t* alebo *b*. Týmto sa riadi vertikálne uloženie boxu vzhľadom na základnú čiaru okolitého textu. to control the vertical alignment of the box, relative to the baseline of the surrounding text. Parameter *šírka* je dĺžkový argument určujúci šírku boxu. Základný rozdiel medzi príkazmi *minipage* a *parbox* je, že vo vnútri *parboxu* nemôžete používať všetky príkazy a prostredia, zatiaľ čo vo vnútri *minipage* je možné takmer všetko.

Zatiaľ čo *\parbox* zabalí celý odstavec so zalomením riadkov, existuje aj trieda príkazov, vytvárajúcich boxy, ktoré pracujú len s horizontálne uloženým materiálom. Už poznáme jeden z nich. Nazýva sa *\mbox* a jednoducho spája do jedného boxu rad boxov. Môže byť použitý na zabránenie L^AT_EXu rozdeliť dve slová. Keďže dovnútra boxov môžete uložiť ďalšie boxy, tieto príkazy na horizontálne balenie boxov vám poskytujú obrovskú flexibilitu.

```
\makebox[šírka][pozícia]{text}
```

šírka určuje šírku výsledného boxu videného zvonku.⁸ Oddelene od dĺžkových výrazov môžete použiť aj *\width*, *\height*, *\depth* a *\totalheight* v parametri šírky. Tieto sa nastavujú podľa hodnôt, získaných meraním vytlačeného *textu*. Parameter *pozícia* nadobúda jednu z hodnôt: *centrovane*, *left flush* (zarovnanie naľavo), *right flush* (zarovnanie vpravo) alebo *s*, ktorá rozťahne text, obsiahnutý v boxe, na celú šírku boxu.

Príkaz *\framebox* funguje presne tak ako *\makebox*, ale navyše nakreslí okolo textu rámček.

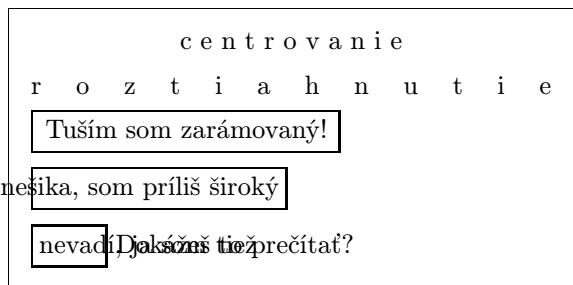
Nasledujúci príklad vám ukáže niektoré veci, ktoré môžete robiť príkazmi *\makebox* a *\framebox*.

⁸To znamená, že môže byť menšia ako materiál, obsiahnutý vo vnútri boxu. Dokonca môžete nastaviť šírku na 0pt, takže text vo vnútri boxu sa vytlačí bez vplyvu na okolité boxy.


```

\makebox[\textwidth]{%
  centrovane}\par
\makebox[\textwidth][s]{%
  roztrahnute}\par
\framebox[1.1\width]{Tuším
  som zarámovaný!} \par
\framebox[0.8\width][r]{Ty nešika, Ty nešika, som príliš široký} \par
\framebox[1cm][l]{nevadí,
  ja som tiež}
Dokážeš to prečítať?

```



Teraz, keď sme zvládli vodorovné riadenie, očividný nasledujúci krok je prechod na vertikálne.⁹ Žiaden problém pre L^AT_EX. Príkazom

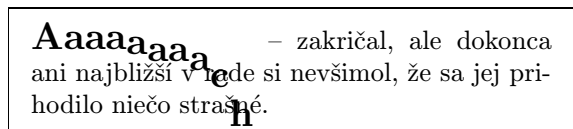
```
\raisebox{nadvihnutie}[hlbka][výška]{text}
```

môžete určiť vertikálne vlastnosti boxu. V prvých troch parametroch môžete použiť `\width`, `\height`, `\depth` a `\totalheight` aby ste ovplyvnili veľkosť boxu pre argument *text*.

```

\raisebox{0pt}[0pt][0pt]{\Large%
\textbf{Aaaa}\raisebox{-0.3ex}{a}%
\raisebox{-0.7ex}{aa}%
\raisebox{-1.2ex}{a}%
\raisebox{-2.2ex}{c}%
\raisebox{-4.5ex}{h}}
-- zakričal, ale dokonca ani
najbližší v rade si nevšimol, že
sa jej prihodilo niečo strašné.

```



5.7 Čiary a rozpery

Pred niekoľkými stranami ste si mohli všimnúť príkaz

```
\rule[nadvihnutie]{šírka}{výška}
```

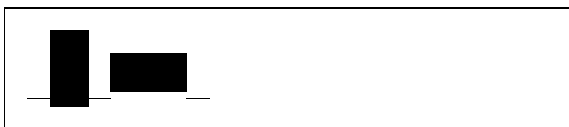
Pri obyčajnom použití vytvorí jednoduchý čierny obdĺžnik.

⁹Úplnú kontrolu nad všetkým môžeme získať len kombináciou horizontálneho a vertikálneho

```

\rule{3mm}{.1pt}%
\rule[-1mm]{5mm}{1cm}%
\rule{3mm}{.1pt}%
\rule[1mm]{1cm}{5mm}%
\rule{3mm}{.1pt}

```



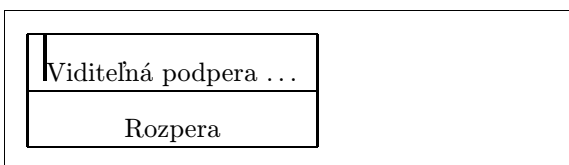
Toto je užitočné na kreslenie vertikálnych a horizontálnych čiar. Napríklad čiara na titulnej strane bola vytvorená príkazom `\rule`.

Zvláštnym prípadom je čiara s nulovou šírkou, ale s určitou výškou. V profesionálnej typografii sa to anglicky nazýva *strut*.¹⁰ Používa sa na zabezpečenie toho, aby prvok stránky mal určitú minimálnu výšku. Môžete ju využiť v prostredí `tabular` na zaistenie určitej minimálnej výšky riadku.

```

\begin{tabular}{|c|}
\hline
\rule{1pt}{4ex}Viditeľná podpera
\ldots\ \hline
\rule{0pt}{4ex}Rozpera\ \hline
\hline
\end{tabular}

```



¹⁰Žiaľ, neviem, aký je slovenský typografický názov pre niečo také. Snáď to vystihuje slovo rozpera. (pozn. prekl.)

Literatúra

- [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] Michel Goossens, Frank Mittelbach and Alexander Samarin. *The L^AT_EX Companion*. Addison-Wesley, Reading, Massachusetts, 1994, ISBN 0-201-54199-8.
- [4] Každá L^AT_EXová inštalácia by mala obsahovať tzv. *L^AT_EX Local Guide* (miestneho sprievodcu programom L^AT_EX), ktorý popisuje špeciality miestneho systému. Mal by sa nachádzať v súbore `local.tex`. Nanešťastie niektoré lenivé systémy tento dokument neposkytujú. V takom prípade musíte o radu poprosiť miestneho L^AT_EXového guru.
- [5] L^AT_EX3 Project Team. *L^AT_EX 2_ε pre autorov*. V inštalácii L^AT_EX 2_ε ako `usrguide.tex`.
- [6] L^AT_EX3 Project Team. *L^AT_EX 2_ε pre autorov Tried a Balíkov*. V inštalácii L^AT_EX 2_ε ako `clsguide.tex`.
- [7] L^AT_EX3 Project Team. *L^AT_EX 2_ε voľba fontov*. V inštalácii L^AT_EX 2_ε ako `fntguide.tex`.
- [8] D. P. Carlisle. *Balíky zo zväzku 'graphics'*. So zväzkom 'graphics' je distribuovaný aj súbor `grfguide.tex`, dostupný z rovnakého zdroja ako inštalácia L^AT_EX.
- [9] Rainer Schöpf, Bernd Raichle, Chris Rowley. *A New Implementation of L^AT_EX's verbatim Environments*. Je distribuovaný so zväzkom 'tools' ako `verbatim.dtx`, dostupný z rovnakého zdroja ako inštalácia L^AT_EX.
- [10] Graham Williams. *The TeX Catalogue* je jeden z najúplnejších zoznamov mnohých balíkov súvisiacich s T_EXom a L^AT_EXom. Dostupný online z CTAN:/tex-archive/help/Catalogue/catalogue.html

- [11] Keith Reckdahl. *Using EPS Graphics in L^AT_EX 2_ε Documents* vysvetľuje všetko a oveľa viac ako by ste si kedykoľvek priali vedieť o EPS-s[boroch a ich použití v L^AT_EXovských dokumentoch. Dostupný online z CTAN:/tex-archive/info/epslatex.ps
- [12] J. Rybička. *L^AT_EX pro začátečníky*. Konvoj, Brno, 1995 (144).

Register

\!, 45
", 21
\$, 39
, 42
\(, 39
\), 39
\,, 40, 45
-, 22
—, 22
\-, 20
—, 22
—, 22
., medzera za, 25
... , 22
.aux, 14
.cls, 13
.dtx, 13
.dvi, 13
.idx, 14
.ilg, 14
.ind, 14
.ins, 13
.lof, 13
.log, 13
.lot, 13
.sty, 13
.tex, 13
.toc, 13
\:, 45
\;, 45
\@, 25
\[, 40
\[, 19, 29, 30, 32, 78
*, 19
\], 40
^, 42
_, 42
|, 32
~, 25
acute, 23
\addtolength, 80
æ, 23
akcenty, 42
align, 48
amsbsy, 51
amsfonts, 41, 58
amsmath, 43, 45, 46, 48, 51
amssymb, 41, 52
\and, 27
\appendix, 26
array, 46
\atop, 43
\author, 27
\authors, 27
babel, 20, 24
\backmatter, 27
balík, 11, 69
balíky, 8
amsbsy, 51
amsfonts, 41, 58
amsmath, 43, 45, 46, 48, 51
amssymb, 41, 52
babel, 20, 24
calc, 80
dcolumn, 33
doc, 12
eucal, 58
eufrak, 58
exscale, 12, 45

- fancyhdr, 65, 66
- fontenc, 12, 25
- graphicx, 59
- ifthen, 12
- indentfirst, 76
- inputenc, 12, 24
- latexsym, 12
- layout, 78
- makeidx, 12, 64
- mathrsfs, 58
- showidx, 65
- syntonly, 12, 15
- tools, 33, 78
- verbatim, 7, 65, 66
- \begin, 29
 - bez bodky i a j, 23
- \bibitem, 63
- \big, 45
- \Big, 45
- \bigg, 45
- \Bigg, 45
- \binom, 43
- \bmod, 43
 - bodka, 22
 - bodky
 - diagonálne, 45
 - horizontálne, 45
 - tri, 45
 - vertikálne, 45
- \boldmath, 51
- \boldsymbol, 51
- calc, 80
- \caption, 36, 66
- \cdots, 45
 - center, 29
- \ci, 69
- \cite, 63
- \cleardoublepage, 37
- \clearpage, 37
- \cline, 32
 - command, 69
 - comment, 7
 - csindex, 64
 - čiarka, 22, 42
- \date, 27
 - dcolumn, 33
- \ddots, 45
 - delenie slov, 20
- \depth, 82, 83
 - desatinné zarovnávanie, 33
 - description, 29
 - diakritika, 23
 - displaymath, 40
- \displaystyle, 44, 49
 - dlhé rovnice, 47
 - doc, 12
- \documentclass, 13
- \documentclass, 8, 9
 - dolný index, 42
 - druhá odmocnina, 42
- \dum, 69
 - dva stĺpce, 10
- em-dash, 22
- \emph, 28, 73
 - empty, 11
 - en-dash, 22
 - ENCAPSULATED POSTSCRIPT, 59
- \end, 29
 - enumerate, 29
 - eqnarray, 47
 - equation, 40
 - eucal, 58
 - eufrak, 58
 - exponent, 42
 - exscale, 12, 45
 - extension, 11
- fancyhdr, 65, 66
- farebný text, 11
- figure, 35, 36
- flushleft, 29
- flushright, 29
- fontenc, 12, 25
- fonty
 - typy, 72, 73
 - veľkosti, 72, 74

- \footnote, 28, 66
- \footnotesize, 74
 - formát stránky (\pagestyle)
 - empty, 11
 - headings, 11
 - plain, 11
- \frac, 43
- \framebox, 82
- \frenchspacing, 25
- \frontmatter, 27
 - funkcie, 43
- \fussy, 20

- \genfrac, 43
 - GHOSTSCRIPT, 59
 - grafika, 11
 - graphicx, 59
 - grave, 23
 - grécke písmená, 41

 - headings, 11
- \height, 82, 83
- \hline, 32
 - horizontálne
 - čiara, 42
 - svorky, 42
 - horný index, 42
 - hranaté zátvorky, 6
- \hspace, 71, 77
- \huge, 74
- \Huge, 74
 - hyphen, 22
- \hyphenation, 20

- \chapter, 26, 65
- \chaptermark, 65
- \choose, 43

- \idotsint, 46
 - ifthen, 12
- \iiiint, 46
- \iiint, 46
- \iint, 46
- \include, 14
- \includegraphics, 60, 81

- \includeonly, 14
- \indent, 76
 - indentfirst, 76
- \index, 64
- \input, 14
 - inputenc, 12, 24
- \int, 44
 - integrál, 44
- \item, 29
 - itemize, 29

- jednotky, 77, 78

- Knuth, Donald E., 1
- kódovanie fontov, 12
- krátka pomlčka, 22
- krehké príkazy, 66
- krížové odkazy, 27

- \label, 27, 40
 - Lamport, Leslie, 2
- \large, 74
- \Large, 74
- \LARGE, 74
- \LaTeX, 21
 - L^AT_EX2.09, 2
 - L^AT_EX 2_ε, 2
 - L^AT_EX3, 2, 4
- \LaTeXe, 21
 - latexsym, 12
 - layout, 78
- \ldots, 22, 45
- \left, 44
- \leftmark, 65
 - ligatúry, 23
- \limits, 44
- \linebreak, 19
- \linespread, 76
- \listoffigures, 36
- \listoftables, 36

- \mainmatter, 27
- \makebox, 82
 - makeidx, 12, 64
 - makeindex, 64

- `\makeindex`, 64
- `\maketitle`, 27
- makrá, 8
- matematika, 39
 - akcenty, 42
 - bodky
 - diagonálne, 45
 - horizontálne, 45
 - tri, 45
 - vertikálne, 45
 - čiarka, 42
 - dlhé rovnice, 47
 - dolný index, 42
 - druhá odmocnina, 42
 - exponent, 42
 - funkcie, 43
 - horizontálne
 - čiara, 42
 - svorky, 42
 - horný index, 42
 - integrál, 44
 - medzery, 45
 - mínus, 22
 - modulo, 43
 - oddeľovač, 45
 - suma, 44
 - sústava rovníc, 47
 - vektory, 42
 - zátvorky, 44
 - zlomky, 43
 - znak šípky, 42
- `math`, 39
- `\mathbb`, 41
- `\mathbf`, 74
- `\mathcal`, 74
- `\mathit`, 74
- `\mathnormal`, 74
- `\mathrm`, 49, 74
 - `mathrsfs`, 58
- `\mathsf`, 74
- `\mathtt`, 74
- `\mbox`, 21, 23, 82
 - medzera
 - horizontálna, 77
 - na začiatku riadku, 5
 - po príkaze, 6
 - vertikálna, 78
 - medzery, 45
 - medzinárodná jazyková podpora, 24
 - `minipage`, 82
 - Mittelbach, Frank, 2
 - mínus, 22
 - modulo, 43
 - `\multicolumn`, 33
 - názov, 10
 - názov dokumentu, 10
 - nemecký jazyk, 24
 - nepovinné parametre, 6
 - `\newcommand`, 70
 - `\newenvironment`, 71
 - `\newline`, 19
 - `\newpage`, 19
 - `\newtheorem`, 49, 50
 - `\noindent`, 77
 - `\nolinebreak`, 19
 - `\nonumber`, 48
 - `\nopagebreak`, 19
 - `\normalsize`, 74
 - obrátene lomítka, 6
 - obsah, 26
 - oddeľovač, 45
 - oddeľovače, 44
 - odstavec, 17
 - œ, 23
 - option, 9
 - `\overbrace`, 42
 - overfull hbox, 20
 - `\overleftarrow`, 42
 - `\overline`, 42
 - `\overrightarrow`, 42
 - page style, 11
 - `\pagebreak`, 19
 - `\pageref`, 27
 - `\pagestyle`, 11
 - papier

- A4, 10
- A5, 10
- B5, 10
- hlavičkový, 10
- legal, 10
- listový, 10
- paragraph, 17
- `\paragraph`, 26
- parameter, 6
- `\parbox`, 82
- parbox, 82
- `\parindent`, 76
- `\parskip`, 76
- `\part`, 26
- päta strany, 11
- `\phantom`, 48, 66
- `\picture`, 61
- plain, 11
- plávající objekty, 35
- `\pmb`, 51
- `\pmod`, 43
- pomlčka, 22
- pomlčky
 - em-dash, 22
 - en-dash, 22
 - krátka pomlčka, 22
 - pomlčka, 22
- POSTSCRIPT, 59
- poznámky, 7
- preambula, 8
- prehláska, 23
- `\printindex`, 65
- příkazy, 6
 - `\!`, 45
 - `\(`, 39
 - `\)`, 39
 - `\,`, 40, 45
 - `\-`, 20
 - `\:`, 45
 - `\;`, 45
 - `\@`, 25
 - `\[`, 40
 - `\`, 19, 29, 30, 32, 78
 - `*`, 19
 - `\]`, 40
 - `\addtolength`, 80
 - `\and`, 27
 - `\appendix`, 26
 - `\atop`, 43
 - `\author`, 27
 - `\authors`, 27
 - `\backmatter`, 27
 - `\begin`, 29
 - `\bibitem`, 63
 - `\big`, 45
 - `\Big`, 45
 - `\bigg`, 45
 - `\Bigg`, 45
 - `\binom`, 43
 - `\bmod`, 43
 - `\boldmath`, 51
 - `\boldsymbol`, 51
 - `\caption`, 36, 66
 - `\cdots`, 45
 - `\ci`, 69
 - `\cite`, 63
 - `\cleardoublepage`, 37
 - `\clearpage`, 37
 - `\cline`, 32
 - `\date`, 27
 - `\ddots`, 45
 - `\depth`, 82, 83
 - `\displaystyle`, 44, 49
 - `\documentclass`, 13
 - `\documentclass`, 8, 9
 - `\dum`, 69
 - `\emph`, 28, 73
 - `\end`, 29
 - `\footnote`, 28, 66
 - `\footnotesize`, 74
 - `\frac`, 43
 - `\framebox`, 82
 - `\frenchspacing`, 25
 - `\frontmatter`, 27
 - `\fussy`, 20
 - `\genfrac`, 43
 - `\height`, 82, 83
 - `\hline`, 32

-
- \hspace, 71, 77
 - \huge, 74
 - \Huge, 74
 - \hyphenation, 20
 - \chapter, 26, 65
 - \chaptermark, 65
 - \choose, 43
 - \idotsint, 46
 - \iiiint, 46
 - \iiint, 46
 - \iint, 46
 - \include, 14
 - \includegraphics, 60, 81
 - \includeonly, 14
 - \indent, 76
 - \index, 64
 - \input, 14
 - \int, 44
 - \item, 29
 - \label, 27, 40
 - \large, 74
 - \Large, 74
 - \LARGE, 74
 - \LaTeX, 21
 - \LaTeXe, 21
 - \ldots, 22, 45
 - \left, 44
 - \leftmark, 65
 - \limits, 44
 - \linebreak, 19
 - \linespread, 76
 - \listoffigures, 36
 - \listoftables, 36
 - \mainmatter, 27
 - \makebox, 82
 - \makeindex, 64
 - \maketitle, 27
 - \mathbb, 41
 - \mathbf, 74
 - \mathcal, 74
 - \mathit, 74
 - \mathnormal, 74
 - \mathrm, 49, 74
 - \mathsf, 74
 - \mathtt, 74
 - \mbox, 21, 23, 82
 - \multicolumn, 33
 - \newcommand, 70
 - \newenvironment, 71
 - \newline, 19
 - \newpage, 19
 - \newtheorem, 49, 50
 - \noindent, 77
 - \nolinebreak, 19
 - \nonumber, 48
 - \nopagebreak, 19
 - \normalsize, 74
 - \overbrace, 42
 - \overleftarrow, 42
 - \overline, 42
 - \overrightarrow, 42
 - \pagebreak, 19
 - \pageref, 27
 - \pagestyle, 11
 - \paragraph, 26
 - \parbox, 82
 - \parindent, 76
 - \parskip, 76
 - \part, 26
 - \phantom, 48, 66
 - \picture, 61
 - \pmb, 51
 - \pmod, 43
 - \printindex, 65
 - \protect, 67
 - \providecommand, 70
 - \ProvidesPackage, 72
 - \qqquad, 40, 45
 - \quad, 40, 45
 - \raisebox, 83
 - \ref, 27, 40
 - \renewcommand, 70
 - \renewenvironment, 71
 - \right, 44, 46
 - \right., 44
 - \rightmark, 65
 - \rule, 71, 83, 84
 - \scriptscriptstyle, 49

- `\scriptsize`, 74
- `\scriptstyle`, 49
- `\section`, 26, 65, 66
- `\sectionmark`, 65
- `\sethlength`, 80
- `\setlength`, 76, 80
- `\settodepth`, 81
- `\settoheight`, 81
- `\settowidth`, 81
- `\sloppy`, 20
- `\small`, 74
- `\sqrt`, 42
- `\stackrel`, 44
- `\stretch`, 71, 77
- `\subparagraph`, 26
- `\subsection`, 26
- `\subsectionmark`, 65
- `\subsubsection`, 26
- `\sum`, 44
- `\tableofcontents`, 26
- `\TeX`, 21
- `\textbf`, 73
- `\textit`, 73
- `\textmd`, 73
- `\textnormal`, 73
- `\textrm`, 49, 73
- `\textsc`, 73
- `\textsf`, 73
- `\textsl`, 73
- `\textstyle`, 44, 49
- `\texttt`, 73
- `\textup`, 73
- `\thispagestyle`, 11
- `\tiny`, 74
- `\title`, 27
- `\today`, 21
- `\totalheight`, 82, 83
- `\underbrace`, 42
- `\underline`, 42
- `\usepackage`, 11, 13, 24, 25, 72
- `\vdots`, 45
- `\vec`, 42
- `\verb`, 31, 32
- `\verbatiminput`, 66
- `\vspace`, 77
- `\widehat`, 42
- `\widetilde`, 42
- `\width`, 82, 83
- program na usporiadanie registra
 - `csindex`, 64
 - `makeindex`, 64
- prostredia, 29
 - `align`, 48
 - `array`, 46
 - `center`, 29
 - `command`, 69
 - `comment`, 7
 - `description`, 29
 - `displaymath`, 40
 - `enumerate`, 29
 - `eqnarray`, 47
 - `equation`, 40
 - `figure`, 35, 36
 - `flushleft`, 29
 - `flushright`, 29
 - `itemize`, 29
 - `math`, 39
 - `minipage`, 82
 - `parbox`, 82
 - `quotation`, 30
 - `quote`, 30
 - `split`, 48
 - `tabbing`, 34
 - `table`, 35, 36
 - `tabular`, 32, 81
 - `thebibliography`, 63
 - `verbatim`, 31, 65, 66
 - `verse`, 30
- `\protect`, 67
- `\providecommand`, 70
- `\ProvidesPackage`, 72
- `\qqquad`, 40, 45
- `\quad`, 40, 45
- `quotation`, 30
- `quote`, 30

- `\raisebox`, 83
- `\ref`, 27, 40
 - register, 64
- `\renewcommand`, 70
- `\renewenvironment`, 71
 - rezervované symboly, 5
 - riadkovanie, 76
 - dvojité, 76
- `\right`, 44, 46
- `\right.`, 44
- `\rightmark`, 65
 - rozdeľovník, 22
 - rozmary, 77
 - rozmary papiera, 10
 - rozmary textu, 78
 - rozvrhnutie stránky, 78
- `\rule`, 71, 83, 84

- `\scriptscriptstyle`, 49
- `\scriptsize`, 74
- `\scriptstyle`, 49
- `\section`, 26, 65, 66
- `\sectionmark`, 65
- `\setlength`, 80
- `\setlength`, 76, 80
- `\settodepth`, 81
- `\settoheight`, 81
- `\settowidth`, 81
 - showidx, 65
- `\sloppy`, 20
- `\small`, 74
 - split, 48
 - spojovník, 22
- `\sqrt`, 42
- `\stackrel`, 44
- `\stretch`, 71, 77
 - strut, 84
- `\subparagraph`, 26
- `\subsection`, 26
- `\subsectionmark`, 65
- `\subsubsection`, 26
- `\sum`, 44
 - suma, 44
 - sústava rovníc, 47
- syntonly, 12, 15

- škandinávské písmená, 23
- špeciálne znaky, 23
- špecifikátor umiestnenia, 35
- štruktúra, 7
- štýl dokumentu (`documentclass`)
 - `article` (článok), 9
 - `book` (kniha), 9
 - `letter` (list), 9
 - `report` (správa), 9
 - `slide` (priesvitka), 9

- tabbing, 34
- table, 35, 36
- `\tableofcontents`, 26
- tabular, 32, 81
- `\TeX`, 21
- `\textbf`, 73
- `\textit`, 73
- `\textmd`, 73
- `\textnormal`, 73
- `\textrm`, 49, 73
- `\textsc`, 73
- `\textsf`, 73
- `\textsl`, 73
- `\textstyle`, 44, 49
- `\texttt`, 73
- `\textup`, 73
 - thebibliography, 63
- `\thispagestyle`, 11
 - tilde, 22
- `\tiny`, 74
- `\title`, 27
 - titul, 27
 - tlač
 - jednostranná, 10
 - obojsstranná, 10
- `\today`, 21
 - tools, 33, 78
- `\totalheight`, 82, 83
 - trieda dokumentu (`documentclass`)
 - `article` (článok), 9
 - `book` (kniha), 9

- letter (list), 9
- report (správa), 9
- slide (priesvitka), 9
- tučné znaky, 41, 51
- typoch súborov, 11
- typy písma
 - antikva (roman), 73
 - bezpätičkové (sans serif), 73
 - blackboard bold, 41
 - kurzíva (italic), 73
 - malé kapitálky (small caps), 73
 - matematická kurzíva, 41
 - naklonené (slanted), 73
 - písací stroj (typewriter), 73
 - polotučné (bold face), 73
 - zvýraznené (emphasised), 73
- umlaut, 23
- `\underbrace`, 42
 - underfull hbox, 20
- `\underline`, 42
 - URL, 22
- `\usepackage`, 11, 13, 24, 25, 72
- úvodzovky, 21
- `\vdots`, 45
- `\vec`, 42
 - vektory, 42
- veľkosť matematického fontu, 48
- veľkosť papiera, 78
- veľkosť písma dokumentu, 10
- `\verb`, 31, 32
 - verbatim, 7, 65, 66
 - verbatim, 31, 65, 66
- `\verbatiminput`, 66
- verse, 30
- vlnka, 22
- vlnovka (~), 25
- `\vspace`, 77
 - výhody L^AT_EXu, 4
- vzorcie, 39
- `\widehat`, 42
- `\widetilde`, 42
- `\width`, 82, 83
 - www, 22
 - WYSIWYG, 3, 4
- záhlavie, 11
- základná veľkosť písma, 10
- zarovnanie doprava, 29
- zarovnanie doľava, 29
- zátvorky, 44
- zdrojový súbor, 8
- zlomky, 43
- zložené zátvorky, 6, 73
- znak mínus, 22
- znak šípky, 42
- zoskupovanie, 73
- zoznam použitej literatúry, 63