

Ne úplně nejkratší úvod do formátu L^AT_EX 2_ε

aneb L^AT_EX 2_ε za 152 minut

autorů **Tobiase Oetikera**

Huberta Partla, Irene Hyna a Elisabeth Schlegl

Verze 4.27, 13. prosince 2009

Překladaatelé: Michal Mádr a Pavel Stříž <striz@fame.utb.cz>

Řadu připomínek poskytli: Miloš Brejcha a Karel Píška

Poslední aktualizace: 15. června 2011

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

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

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

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

Tento dokument je volně k dispozici; můžete ho dále šířit a/nebo modifikovat v souladu s GNU General Public License publikovanou organizací Free Software Foundation (verze 2 nebo novější).

Tento dokument je šířen v naději, že bude užitečný, ale BEZ JAKÉKOLIV ZÁRUKY; bez implicitní záruky OBCHODOVATELNOSTI nebo VHODNOSTI PRO KONKRÉTNÍ ÚČEL. Více podrobností najdete v textu GNU General Public License.

Spolu s tímto dokumentem byste měli obdržet i GNU General Public License. Pokud tomu tak není, kontaktujte Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

Děkujeme!

Velká část materiálů použitých v tomto úvodu pochází z rakouského úvodu do L^AT_EXu 2.09 autorů:

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

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

Elisabeth Schlegl <noemail>
in Graz

Máte-li zájem o německou verzi tohoto dokumentu, kterou Jörg Knappen aktualizoval pro L^AT_EX 2_ε, najdete ji na
CTAN://info/lshort/german

Český překlad

Překlad není nikdy perfektní, proto předem děkujeme za případné připomínky, návrhy a komentáře.

Michal Mádr, Pavel Stříž
<striz@fame.utb.cz>

Pavel Stříž
Tomas Bata University in Zlín
Faculty of Management and Economics
Nám. T. G. Masaryka 5555
76001 Zlín, Czech Republic

Následující jednotlivci pomáhali s korekturami, návrhy a materiály k vylepšení tohoto dokumentu. Vynaložili velké úsilí, aby mi pomohli dostat tento dokument do současného stavu. Chtěl bych jim všem poděkovat. Všechny chyby, na které v dokumentu narazíte, jsou přirozeně moje. A naopak, slova napsaná bez překlepů jsou téměř určitě zásluhou někoho z lidí uvedených níže, kteří mě opravili.

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

Předmluva

L^AT_EX [1] je sázecí systém, který se výborně hodí na tvorbu vědeckých a matematických dokumentů vysoké typografické kvality. L^AT_EX se ale hodí i na tvorbu všech možných dalších typů dokumentů, od jednoduchých dopisů po knihy. L^AT_EX používá T_EX [2] jako svůj formátovací stroj.

Tento krátký úvod popisuje L^AT_EX 2_ε a měl by pokrývat většinu aplikací L^AT_EXu. Úplný popis systému L^AT_EX naleznete v [1, 3].

Tento úvod je rozdělen do šesti kapitol:

Kapitola 1 vás seznámí se základní strukturou dokumentů v L^AT_EX 2_ε. Dozvíte se také něco o historii L^AT_EXu. Po přečtení této kapitoly byste měli zhruba rozumět tomu, jak L^AT_EX funguje.

Kapitola 2 popisuje detaily sazby vašeho dokumentu. Vysvětluje většinu základních L^AT_EXových příkazů a prostředí. Po přečtení této kapitoly budete schopni napsat své první dokumenty.

Kapitola 3 vysvětluje, jak se v L^AT_EXu sází matematické vzorce. Řada příkladů ukazuje, jak se tato jedna z největších předností L^AT_EXu používá. Na konci kapitoly jsou tabulky všech matematických symbolů dostupných v L^AT_EXu.

Kapitola 4 vysvětluje indexy, tvorbu bibliografií a vkládání EPS grafiky. Dále představuje tvorbu PDF dokumentů pomocí pdfL^AT_EXu a některé užitečné balíky.

Kapitola 5 ukazuje, jak lze pomocí L^AT_EXu vytvářet grafiku jinak, než „normálním“ způsobem (místo toho, abyste obrázek nakreslili v některém z grafických programů, uložíte do externího souboru a vložíte do L^AT_EXového dokumentu, vložíte do souboru s textem L^AT_EXového dokumentu jeho popis a necháte L^AT_EX, aby vám podle něj obrázek nakreslil).

Kapitola 6 obsahuje potenciálně nebezpečné informace týkající se toho, jak lze změnit standardní vzhled L^AT_EXového dokumentu. Přečtete si o tom, jak udělat změny, které krásný vzhled L^AT_EXových dokumentů mohou zlepšit ještě víc, ale mohou ho také úplně pokazit.

Kapitoly je potřeba číst od první do poslední – dokument nakonec není tak rozsáhlý. Mnoho informací je obsaženo v příkladech a bude proto nejlepší, když si je důkladně projdete.

L^AT_EX je dostupný na většině počítačů, od PC a Maců po velké systémy typu UNIX a VMS. Na počítačích mnoha univerzitních sítí je L^AT_EX nainstalován a připraven k použití. Informace o tom, jak získat přístup k lokální L^AT_EXové instalaci, by měla být dostupná v *Local Guide* [5]. Budete-li mít problémy s prvními kroky, obraťte se na osobu, od které jste tuto brožurku získali. Tento dokument vás *nenaučí*, jak nainstalovat a zprovoznit systém L^AT_EX, jeho cílem je naučit vás, jak napsat dokumenty, které bude možno L^AT_EXem zpracovat.

Hledáte-li jakékoliv materiály týkající se L^AT_EXu, navštivte jedno z míst, kde je uložen Comprehensive T_EX Archive Network (CTAN – Síť obsáhlých T_EXových archivů). Začít můžete na adrese <http://www.ctan.org>. Všechny balíky lze také získat z ftp archivu <ftp://www.ctan.org> a jeho „zrcadel“ umístěných po celém světě.

V dokumentu budete narážet na odkazy do CTANu, zvláště na software a dokumenty, které si můžete stáhnout. Místo uvádění kompletních URL vždy píšu jen CTAN: následované lokací uvnitř stromu CTANu, kam se vydat.

Chcete-li L^AT_EX používat na vašem vlastním počítači, poohlédněte se, co je dostupné na <CTAN://tex-archive/systems>.

Máte-li nápady, co by se do dokumentu mohlo přidat, odebrat nebo změnit, dejte mi prosím vědět. Zvláště by mě zajímaly reakce od L^AT_EXových začátečníků – které části tohoto dokumentu jsou dobře pochopitelné a které by mohly být napsány lépe.

Tobias Oetiker <tobi@oetiker.ch>

OETIKER+PARTNER AG
Aarweg 15
4600 Olten
Switzerland

Nejnovější verze tohoto dokumentu je dostupná na (český překlad):
<CTAN://lshort>

Obsah

Děkujeme!	iii
Předmluva	v
1. Věci, které je dobré vědět	1
1.1. Jméno hry aneb oč tu běží	1
1.1.1. T _E X	1
1.1.2. L ^A T _E X	1
1.2. Základy	2
1.2.1. Autor, typograf a sazeč	2
1.2.2. Layout	2
1.2.3. Výhody a nevýhody	3
1.3. Vstupní soubory L ^A T _E Xu	4
1.3.1. Mezery	4
1.3.2. Speciální znaky	4
1.3.3. L ^A T _E Xové příkazy	5
1.3.4. Komentáře	6
1.4. Struktura vstupního souboru	6
1.5. Typické použití z příkazové řádky	7
1.6. Layout dokumentu	9
1.6.1. Třídy dokumentů	9
1.6.2. Balíky	11
1.6.3. Stránkové styly	11
1.7. Soubory, na které můžete narazit	11
1.8. Velké projekty	14
2. Sazba textu	15
2.1. Struktura textu a jazyka	15
2.2. Řádkový a stránkový zlom	17
2.2.1. Zarovnané odstavce	17
2.2.2. Dělení slov	18
2.3. Předpřipravené řetězce	19
2.4. Zvláštní znaky a symboly	19

2.4.1.	Uvozovky	19
2.4.2.	Pomlčky a spojovníky	20
2.4.3.	Tilda (\sim)	20
2.4.4.	Symbol stupně (\circ)	20
2.4.5.	Symbol Eura (€)	21
2.4.6.	Výpustky (\dots)	21
2.4.7.	Ligatury	22
2.4.8.	Akcenty a speciální znaky	22
2.5.	Podpora pro neanglické jazyky	22
2.5.1.	Podpora pro portugalštinu	25
2.5.2.	Podpora pro francouzštinu	26
2.5.3.	Podpora pro němčinu	27
2.5.4.	Podpora pro korejštinu	27
2.5.5.	Sazba řečtiny	30
2.5.6.	Podpora cyriliky	30
2.5.7.	Podpora pro mongolštinu	31
2.6.	Mezery mezi slovy	33
2.7.	Titulky, kapitoly, sekce	33
2.8.	Křížové odkazy	35
2.9.	Poznámky pod čarou	36
2.10.	Zdůrazněná slova	36
2.11.	Prostředí	37
2.11.1.	Itemize, Enumerate a Description	37
2.11.2.	Flushleft, Flushright a Center	38
2.11.3.	Citace, citáty a verše	39
2.11.4.	Abstrakty	39
2.11.5.	Sazba doslovně	40
2.11.6.	Tabular	40
2.12.	Plovoucí objekty	43
2.13.	Chránění „zranitelných“ příkazů	46
3.	Sazba matematických vzorců	49
3.1.	Kolekce $\mathcal{AMS-LATEX}$	49
3.2.	Jednoduché rovnice	49
3.2.1.	Matematický mód	51
3.3.	Stavební bloky matematických vzorců	52
3.4.	Příliš dlouhé rovnice: multiline	57
3.5.	Více rovnic	58
3.5.1.	Problémy s tradičními příkazy	59
3.5.2.	Prostředí IEEEeqnarray	60
3.5.3.	Běžné použití	61
3.6.	Pole a matice	64
3.7.	Mezery v matematickém módu	65
3.7.1.	Neviditelné výrazy	66

3.8. Hrátky s matematickými fonty	67
3.8.1. Tučné symboly	67
3.9. Věty, lemmata,	68
3.9.1. Důkazy a symbol „konec důkazu“	69
3.10. Seznam matematických symbolů	72
4. Speciality	81
4.1. Vkládání Encapsulated Postscript grafiky	81
4.2. Bibliografie	83
4.3. Indexování	84
4.4. Efektní záhlaví	85
4.5. Balík Verbatim	87
4.6. Instalace dodatečných balíků	87
4.7. Pracujeme s pdfL ^A T _E Xem	88
4.7.1. PDF dokumenty pro WWW	89
4.7.2. Fonty	89
4.7.3. Používání grafiky	91
4.7.4. Hypertextové odkazy	92
4.7.5. Problémy s odkazy	94
4.7.6. Problémy s bookmarky	94
4.8. Vytváření prezentací	96
5. Tvorba matematické grafiky	101
5.1. Úvodní přehled	101
5.2. Prostředí <code>picture</code>	102
5.2.1. Základní příkazy	102
5.2.2. Řádkové segmenty	103
5.2.3. Šipky	104
5.2.4. Kružnice	105
5.2.5. Text a vzorce	106
5.2.6. <code>\multiput</code> a <code>\linethickness</code>	106
5.2.7. Ovály	107
5.2.8. Vícenásobné použití předdefinovaných boxů s obrázky	108
5.2.9. Kvadratické Bézierovy křivky	109
5.2.10. Řetězovka	110
5.2.11. Rychlost ve speciální teorii relativity	111
5.3. Grafický balík <code>TikZ & PGF</code>	111
6. Přizpůsobování L^AT_EXu	115
6.1. Nové příkazy, prostředí a balíky	115
6.1.1. Nové příkazy	116
6.1.2. Nová prostředí	117
6.1.3. Mezery navíc	117
6.1.4. Příkazová řádka L ^A T _E Xu	118

6.1.5. Váš vlastní balík	118
6.2. Fonty a velikosti	119
6.2.1. Příkazy na změnu fontů	119
6.2.2. Pozor, pozor	122
6.2.3. Rada	123
6.3. Mezery	123
6.3.1. Řádkové mezery	123
6.3.2. Formátování odstavce	123
6.3.3. Horizontální mezery	124
6.3.4. Vertikální mezery	125
6.4. Layout stránky	126
6.5. Další hrátky s délkami	128
6.6. Boxy	129
6.7. Linky	131
A. Instalace L^AT_EXu	133
A.1. Co instalovat	133
A.2. T _E X na Mac OS X	134
A.2.1. Volba editoru	134
A.2.2. Obstarejte si T _E Xovou distribuci	134
A.2.3. Dopřejte si PDFView	134
A.3. T _E X na Windows	134
A.3.1. Obstarání T _E Xu	134
A.3.2. L ^A T _E Xový editor	135
A.3.3. Práce s grafikou	135
A.4. T _E X na Linuxu	135
A.5. Projekt T _E XonWeb	136
Literatura	137
Index	139

Seznam obrázků

1.1	Minimální L ^A T _E Xový soubor.	7
1.2	Příklad reálného článku z časopisu.	7
4.1	Ukázka nastavení fancyhdr.	86
4.2	Ukázkový kód použití třídy beamer.	97
6.1	Pokusný balík.	119
6.2	Parametry rozložení stránky.	127

Seznam tabulek

1.1	Třídy dokumentů.	9
1.2	Nastavení tříd dokumentů.	10
1.3	Některé z balíků distribuovaných s \LaTeX em.	12
1.4	Předdefinované stránkové styly \LaTeX u.	12
2.1	Balík plný symbolů Eura.	21
2.2	Akcenty a speciální znaky.	23
2.3	Preamble pro dokumenty v portugalsštině.	26
2.4	Speciální příkazy pro francouzštinu.	26
2.5	Speciální německé znaky.	27
2.6	Preamble pro řecké dokumenty.	30
2.7	Greek Special Characters.	30
2.8	Bulharština, ruština a ukrajinština.	32
2.9	Povolenky pro umístění plovoucího objektu.	44
3.1	Akcenty matematického módu.	72
3.2	Řecká písmena.	72
3.3	Binární relace.	73
3.4	Binární operátory.	73
3.5	VELKÉ operátory.	74
3.6	Šipky.	74
3.7	Šipky a akcenty.	74
3.8	Oddělovače.	75
3.9	Velké oddělovače.	75
3.10	Různé symboly.	75
3.11	Nematematické symboly.	75
3.12	\mathcal{AMS} oddělovače.	76
3.13	\mathcal{AMS} Řečtina a Hebrejština.	76
3.14	Matematické abecedy.	76
3.15	\mathcal{AMS} Binární operátory.	76
3.16	\mathcal{AMS} Binární relace.	77
3.17	\mathcal{AMS} Šipky.	78
3.18	\mathcal{AMS} Znegované binární relace a šipky.	79

3.19	<i>AMS</i> Různé.	79
4.1	Význam jmen klíčů balíku <i>graphicx</i>	82
4.2	Příklady syntaxe klíčů indexů.	84
6.1	Fonty.	120
6.2	Velikosti fontů.	120
6.3	Absolutní bodové velikosti ve standardních třídách.	121
6.4	Matematické fonty.	121
6.5	<i>TEX</i> ové jednotky.	125

Kapitola 1.

Věci, které je dobré vědět

První část této kapitoly předkládá krátký přehled filozofie a historie systému $\LaTeX 2_{\epsilon}$. Druhá část se soustřeďuje na základní strukturu dokumentu tohoto systému. Po dočtení kapitoly byste zhruba měli vědět, jak \LaTeX funguje. Ve zbytku knihy se předpokládá obeznámenost s touto kapitolou.

1.1. Jméno hry aneb oč tu běží

1.1.1. \TeX

\TeX je počítačový program vytvořený Donaldem E. Knuthem [2]. Úkolem tohoto programu je sázet text a matematické vzorce. Knuth začal \TeX vyvíjet v roce 1977 s cílem prozkoumat potenciál zařízení pro digitální tisk, která v té době začala pronikat do vydavatelského oboru. Knuth chtěl obzvláště zastavit trend klesající kvality typografie, který ovlivnil jeho vlastní knihy a články. \TeX v dnešní podobě byl dokončen v roce 1982, přičemž v roce 1989 byla zlepšena podpora 8-bitových znaků a sazby více jazyků. \TeX má pověst velmi stabilního systému fungujícího na mnoha odlišných typech počítačů, který je v podstatě bez chyb. Číslo verzí \TeX u konvertují k π , aktuální verze je 3.141592.

\TeX vyslovujeme „tech“, což má původ v řečtině ($\tau\epsilon\chi$), kde „X“ je písmenem „ch“ nebo „chi“. \TeX je zároveň první slabikou řeckého slova *texnologia* (technologie). Ve světě ASCII se ze zápisu \TeX stává **TeX**.

1.1.2. \LaTeX

\LaTeX umožňuje autorům sázet a tisknout svá díla v té nejvyšší typografické kvalitě, s použitím předdefinovaných profesionálních layoutů. \LaTeX byl původně napsán Leslieem Lamportem [1]. Používá \TeX jako svůj sázecí stroj. Dnes je \LaTeX udržován Frankem Mittelbachem.

\LaTeX vyslovujeme „lej tech“ nebo „lá tech“. Pomocí ASCII znaků zapisujeme \LaTeX jako `LaTeX`. $\LaTeX 2_{\epsilon}$ vyslovujeme „lej-tech tů í“ a zapisujeme jako `LaTeX2e`.

1.2. Základy

1.2.1. Autor, typograf a sazeč

Tvorba dokumentu začíná u autora, který vytvoří rukopis a předá ho do nakladatelství. Tam se rukopisu ujme typograf a navrhne rozložení stránek dokumentu (šířku sloupců, fonty, velikosti mezer před a za záhlavími, ...). Typograf přepíše příslušné instrukce do rukopisu a ten předá sazeči, který podle nich knihu vysází.

Lidský knižní typograf se snaží přizpůsobit formu výsledného dokumentu obsahu rukopisu, přičemž samozřejmě vychází ze svých profesních znalostí.

\LaTeX působí v roli typografa a používá služeb \TeX u – „sazeče“. Na rozdíl od lidského typografa potřebuje od nás – autorů rukopisu – \LaTeX více instrukcí. Musíme popsat logickou strukturu díla. Tato informace se zapisuje do textu dokumentu pomocí \LaTeX ových příkazů.

To je podstatný rozdíl oproti přístupu, tzv. WYSIWYG¹, který používá většina moderních textových procesorů, např. *Microsoft Word*, *OpenOffice.org Writer* nebo *Corel WordPerfect*. Při práci s těmito programy autor vytváří výsledný dokument interaktivně a na obrazovce v průběhu editace vidí, jak bude vytištěný výsledek vypadat.

Naproti tomu při psaní \LaTeX ového dokumentu na obrazovce přímo nevidíte jeho výslednou podobu. Tu si ale můžete zobrazit po zpracování souboru \LaTeX em. Opravy je tedy možné udělat dříve, než je dokument vytištěn.

1.2.2. Layout

Typografický design je řemeslo. Neškolení autoři často dělají vážné formátovací chyby, když vycházejí z domněnky, že knižní design je hlavně otázka estetiky – „dokument vypadá umělecky, tedy je dobře navržen“. Dokument má ale samozřejmě jiný účel než být pověšen na zeď v galerii. Podstatné je, aby se dobře četl a dobře se v něm orientovalo. Příklad:

- Je potřeba vybrat velikost fontu a číslování nadpisů, aby měl čtenář přehled o struktuře kapitol a sekcí.
- Řádek musí být dostatečně krátký, aby čtenáře neunavovalo přeskakování z konce jednoho na začátek následujícího řádku; zároveň by měla být dost dlouhá na to, aby byla stránka zaplněna.

¹What You See Is What You Get (vytiskne se přesně to, co vidíte na monitoru).

S WYSIWYG systémy autoři často vytvářejí dobře vypadající dokumenty, které ale mají minimální nebo dokonce nejednotnou strukturu. \LaTeX takovým formátovacím chybám předchází tím, že nutí autory explicitně značit *logickou* strukturu svých dokumentů. \LaTeX potom na základě informace o struktuře daného dokumentu zvolí vhodný layout.

1.2.3. Výhody a nevýhody

Když „WYSIWYG lidé“ potkají uživatele \LaTeX u, často se mluví o „výhodách a nevýhodách \LaTeX u“ oproti normálnímu textovému procesoru. Je lepší se v takových chvílích držet zpátky, protože podobné diskuze se často zvrhnou. Někdy ale není vyhnutí. . .

Zde je trocha munice. Hlavní výhody \LaTeX u oproti normálnímu textovému procesoru jsou tyto:

- K dispozici jsou vytvořené layouts, pomocí kterých lze vytvořit dokument, jako kdyby byl vysázen profesionálně.
- Komfortní podpora sazby matematických vzorců.
- Uživateli stačí naučit se pár snadno zapamatovatelných příkazů specifikujících logickou strukturu dokumentu. Téměř nikdy není potřeba zabývat se přímo layoutem dokumentu.
- Dokonce i složité prvky (např. poznámky pod čarou, odkazy, obsahy a bibliografie) lze vytvářet jednoduše.
- K dispozici jsou přídatné balíky řešící řadu typografických úkolů, jejichž řešení není součástí základního \LaTeX u. Lze např. použít balíky pro vkládání obrázků ve formátu POSTSCRIPT nebo pro sazbu bibliografií vyhovujících přesným standardům. Mnoho z těchto přídatných balíků je popsáno v *The \LaTeX Companion* [3].
- \LaTeX vede autory k psaní dobře strukturovaných textů – specifikace struktury je hlavní částí přípravy textu pro zpracování v \LaTeX u.
- \TeX , nízkoúrovňový formátovací nástroj použitý systémem $\LaTeX 2_{\epsilon}$, je snadno přenositelný na nové hardwarové platformy a je zadarmo. Je tedy k dispozici téměř všude.

\LaTeX má také nevýhody:²

- \LaTeX není vhodný pro lidi, kteří prodali své duše. . .

²Já sice o žádných nevím, ale jiní by vám určitě vyjmenovali stovky. ; -)

- Ačkoliv změny některých parametrů v rámci předdefinovaných layoutů dokumentů můžete dělat snadno, navrhnout kompletní nový layout je obtížné a časově náročné.³
- Nestrukturované a neorganizované dokumenty se píšou velmi obtížně.
- Je možné, že navzdory několika počátečním úspěšným pokusům, si na koncept logického značkování nikdy nezvyknete.

1.3. Vstupní soubory L^AT_EXu

Vstupem pro L^AT_EX je obyčejný textový soubor, který můžete vytvořit v libovolném textovém editoru. Tento soubor obsahuje text dokumentu spolu s příkazy, které říkají L^AT_EXu, jak má být text vysázen.

1.3.1. Mezery

S „bílymi“ znaky, např. mezerou nebo tabulátorem, L^AT_EX zachází jednotně jako s „mezerou“. S *několika* bílými znaky za sebou se zachází jako s *jednou* mezerou. Bílý znak na začátku řádku se většinou ignoruje a s koncem řádku se většinou zachází jako s jedním bílým znakem.

Prázdná řádka mezi dvěma řádkami textu značí konec odstavce. S *několika* prázdnými řádky se zachází jako s *jednou*. V příkladu níže je vlevo zobrazen text ze vstupního souboru a na pravé straně příslušný naformátovaný výstup.

Nezáleží na tom, jestli
za slovem napíšete jednu
nebo několik mezer.

Prázdný řádek začíná nový
odstavec.

Nezáleží na tom, jestli za slovem napíšete
jednu nebo několik mezer.

Prázdný řádek začíná nový odstavec.

1.3.2. Speciální znaky

Následující symboly jsou rezervovanými znaky, které buď mají v L^AT_EXu zvláštní význam, nebo nejsou dostupné ve všech fontech. Pokud je uvedete ve vstupním souboru, ve většině případů tím nedocílíte jejich vysázení ve výsledném dokumentu; pravděpodobně tím naopak instruujete L^AT_EX k něčemu, co jste nezamýšleli.

\$ % ^ & _ { } ~ \

³Prosíchlá se v T_EXových kruzích, že toto je jedna z hlavních věcí řešených v připravovaném systému L^AT_EX3.

Jak uvidíte, lze tyto znaky přece jen do vašeho dokumentu vysázet, napíšete-li před ně zpětné lomítko:

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

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

Zbylé symboly a mnohem více lze vytisknout pomocí speciálních příkazů v matematických vzorcích nebo jako akcenty. Znak zpětné lomítka (\) ale *nelze* přidáním dalšího zpětného lomítka zapsat (\\); tato sekvence znaků je totiž použita pro ukončení řádku. Můžete ale použít příkaz `\textbackslash`.

1.3.3. L^AT_EXové příkazy

Ve jménech L^AT_EXových příkazů záleží na tom, jestli písmeno je velké nebo malé. Příkazy mají jednu ze dvou forem:

- Začínají zpětným lomítkem \ následovaným písmeny. První mezera, číslo nebo libovolný jiný „nepísmenný“ znak už netvoří jméno příkazu.
- Začínají zpětným lomítkem následovaným *jedním* nepísmenným znakem.

L^AT_EX ignoruje bílé znaky za jmény příkazů. Pokud chcete, aby bílé znaky následující za jménem L^AT_EXového příkazu byly do vysázeného dokumentu opravdu vloženy, buď za jméno příkazu napíšete {} následované mezerou, nebo použijete speciální příkaz produkující mezeru, např. `_`. Sekvence {} řekne L^AT_EXu, aby neignoroval následující mezery.

```
Četl jsem, že Knuth dělí
lidi pracující s~programem
\TeX{} na \TeX{}niky
a~\TeX perty.\\
Dnes je \today.
```

```
Četl jsem, že Knuth dělí lidi pra-
cující s programem TEX na TEXniky
a TEXperty.
Dnes je 15. června 2011.
```

Některým příkazům je potřeba předat parametr, který se uvádí mezi složené závorky { } následující za jménem příkazu. Některé příkazy podporují nepovinné parametry, které se uvádí mezi hranaté závorky []. Následující příklady používají L^AT_EXové příkazy. Nedělejte si z nich hlavu, později budou vysvětleny.

```
Můžete mě \textsl{nahnout}!
```

```
Můžete mě nahnout!
```

```
Začněte, prosím, novou řádku
právě tady!\newline
Děkuji!
```

```
Začněte, prosím, novou řádku právě tady!
Děkuji!
```

1.3.4. Komentáře

Když \LaTeX při zpracování vstupního souboru narazí na znak %, ignoruje všechny následující znaky až do konce daného řádku a všechny bílé znaky na začátku řádku následujícího.

Díky tomu je možno do vstupních souborů psát komentáře, které ve výsledném dokumentu nebudou vidět.

```
Toto je % stupidní
% nebo lépe: instruktivní <----
příklad: Supercal%
         ifragilist%
         icexpialidocious
```

Toto je příklad: Supercalifragilisticexpialidocious

Znak % lze také použít k rozdělení dlouhých řádků, ve kterých nejsou žádné bílé znaky ani dovolená místa zlomu.

Pro delší komentáře můžete použít prostředí `comment` z balíku `verbatim`. Jak je vysvětleno níže, do preambule vašeho dokumentu musíte přidat řádku `\usepackage{verbatim}`, abyste příkaz `comment` mohli použít.

```
Toto je další
\begin{comment}
celkem stupidní,
ale přesto užitečný
\end{comment}
příklad pro vložení
komentářů do vašich dokumentů.
```

Toto je další příklad pro vložení komentářů do vašich dokumentů.

Prostředí `comment` ale nebude fungovat uvnitř složitých prostředí, např. v prostředí `math`.

1.4. Struktura vstupního souboru

\LaTeX 2_ε u souboru, který zpracovává, očekává určitou strukturu. Konkrétně každý vstupní soubor musí začínat příkazem

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

Tímto specifikujete, jaký druh dokumentu vytváříte. Následují příkazy ovlivňující styl celého dokumentu nebo nahrávající balíky, které přidávají novou funkcionalitu k základnímu \LaTeX u. Takový balík nahrajete příkazem

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

Po provedení všech nastavení⁴ začíná tělo dokumentu příkazem

```
\begin{document}
```

⁴Část dokumentu mezi `\documentclass` a `\begin{document}` se nazývá *preambule*.

Za tento příkaz napíšete text dokumentu spolu s potřebnými L^AT_EXovými příkazy. Na konci dokumentu napíšete příkaz

```
\end{document}
```

který L^AT_EXu říká, aby skončil. Cokoliv následuje za tímto příkazem bude L^AT_EXem ignorováno.

Obrázek 1.1 ukazuje obsah minimálního vstupního souboru pro L^AT_EX 2_ε. Trochu rozsáhlejší vstupní soubor je ukázán na obrázku 1.2.

1.5. Typické použití z příkazové řádky

Určitě už byste si rádi vyzkoušeli zpracovat malý L^AT_EXový vstupní soubor ukázaný na straně 7. Následuje stručný návod: L^AT_EX se dodává bez GUI (grafického uživatelského rozhraní). Je to „jen“ program, který na pozadí zpracuje váš vstupní soubor. *Některé L^AT_EXové instalace obsahují GUI, kde*

```
\documentclass{article}
\begin{document}
Small is beautiful.
\end{document}
```

Obrázek 1.1: Minimální L^AT_EXový soubor.

```
\documentclass[a4paper,11pt]{article}
% define the title
\author{H.~Partl}
\title{Minimalism}
\begin{document}
% generates the title
\maketitle
% insert the table of contents
\tableofcontents
\section{Nějaká zajímavá slova}
A~tady začíná náš článek.
\section{Sbohem, světe}
\ldots{} a~tady končí.
\end{document}
```

Obrázek 1.2: Příklad reálného článku z časopisu. Všechny příkazy, které v příkladu vidíte, jsou vysvětleny níže.

je tlačítko, kterým zpracování svého vstupního souboru odstartujete. Na jiných instalacích je ale nutné odstartování provést z klávesnice, což nyní popíšeme. Upozorněme ještě, že předpokládáme, že máte L^AT_EX na svém počítači už nainstalován.⁵

1. Upravte/vytvořte svůj L^AT_EXový vstupní soubor. Tento soubor musí obsahovat jen obyčejný text. Všechny UNIXové editory vytvoří data právě v tomto formátu. Na Windows budete možná muset explicitně zajistit, aby se soubor uložil ve formátu ASCII nebo *plain text*. Jméno vašeho souboru by mělo mít příponu `.tex`.
2. Spusťte L^AT_EX na svůj vstupní soubor.

```
latex foo.tex
```

Pokud vše dopadne dobře, získáte soubor `.dvi`. Je možné, že bude třeba spustit L^AT_EX víckrát, aby se obsah a vnitřní odkazy vygenerovaly správně. Pokud váš vstupní soubor obsahuje chybu, L^AT_EX vám to oznámí a ve zpracování vašeho souboru přestane. Stiskněte `Ctrl-D`, abyste opět mohli zadávat příkazy z příkazové řádky.

3. Nyní si můžete prohlédnout DVI soubor. Máte několik možností, jak na to. Můžete si soubor prohlédnout na obrazovce pomocí

```
xdvi foo.dvi &
```

Toto bude fungovat jen na Unixu s X11. Na Windows můžete zkusit jiný zobrazovací program, `yap`.

Zkonvertujte `dvi` soubor do formátu `POSTSCRIPT`, abyste si ho mohli vytisknout nebo prohlédnout v programu `Ghostscript`.

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

Možná, že v rámci vašeho L^AT_EXového systému dokonce máte nainstalován program `dvipdf`, který umožňuje konvertovat `.dvi` soubory přímo do `pdf`.

```
dvipdf foo.dvi
```

⁵Tak tomu je téměř u všech Unixových systémů.

1.6. Layout dokumentu

1.6.1. Třídy dokumentů

První informaci, kterou L^AT_EX potřebuje při zpracování vstupního souboru vědět, je typ dokumentu, který chce autor vytvářet. Typ dokumentu se specifikuje pomocí příkazu `\documentclass`.

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

class (třída) specifikuje typ dokumentu, který se má vytvořit. Tabulka 1.1 ukazuje třídy dokumentů rozebrané v tomto úvodu. Distribuce L^AT_EX 2_ε poskytuje přídavné třídy pro další dokumenty, včetně dopisu a slajdů. Parametr *options* upravuje chování třídy dokumentu. Jednotlivé options (volby) musí být odděleny čárkami. Nejčastější volby pro standardní třídy dokumentů jsou uvedeny v tabulce 1.2.

Příklad: vstupní soubor pro L^AT_EX může začínat řádkem

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

který L^AT_EXu říká, aby vysázel dokument jako článek (*article*) se základním fontem o velikosti *jedenáct bodů* a aby použil rozložení stránek vhodné pro *dvoustranný* tisk na papír *velikosti A4*.

Tabulka 1.1: Třídy dokumentů.

article pro články do vědeckých časopisů, prezentace, krátké zprávy, dokumentaci k programům, pozvánky, ...

proc třída pro zápisy založené na třídě article.

minimal má minimální velikost. Umí jen nastavit velikost stránky a základní font. Používá se hlavně pro ladění.

report pro delší zprávy obsahující několik kapitol, pro malé knihy, dizertační práce...

book pro opravdové knihy.

slides pro slajdy. Tato třída používá velká bezserifová písmena. Lepší alternativou může být třída **Beamer**.

Tabulka 1.2: Nastavení tříd dokumentů.

<code>10pt, 11pt, 12pt</code>	Nastaví velikost hlavního fontu dokumentu. Implicitní hodnotou je <code>10pt</code> .
<code>a4paper, letterpaper, ...</code>	Definuje velikost papíru. Implicitní hodnotou je <code>letterpaper</code> . Kromě ní a <code>a4paper</code> lze specifikovat hodnoty <code>a5paper, b5paper, executivepaper</code> a <code>legalpaper</code> .
<code>fleqn</code>	Sází rovnice ve stylu <code>display</code> zarovnané doleva (namísto centrovane).
<code>leqno</code>	Umístí čísla rovnic doleva (namísto doprava).
<code>titlepage, notitlepage</code>	Specifikuje, jestli se má za titulem dokumentu přejít na novou stranu. Třída <code>article</code> implicitně na novou stranu nepřechází, zatímco třídy <code>report</code> a <code>book</code> ano.
<code>onecolumn, twocolumn</code>	Říká \LaTeX u, aby dokument vysázel v jednom, resp. ve více sloupcích.
<code>twoside, oneside</code>	Specifikuje, jestli výstup má být jedno- nebo oboustranný. Třídy <code>article</code> a <code>report</code> jsou implicitně jednostranné, zatímco třída <code>book</code> oboustranná. Upřesněme, že tohle nastavení se týká pouze stylu dokumentu – nastavení <code>twoside</code> <i>neřídí</i> tiskárně, aby papír potiskla z obou stran.
<code>landscape</code>	Změní rozvržení dokumentu tak, že se bude tisknout v módu <code>landscape</code> (po délce papíru, papír otočený o 90 stupňů).
<code>openright, openany</code>	Zařídí, aby kapitoly začínaly vždy na první následující pravé stránce (<code>openright</code>), resp. na první následující stránce (<code>openany</code>). S třídou <code>article</code> toto nefunguje, protože <code>article</code> nepoužívá kapitoly. Třída <code>report</code> implicitně začíná kapitoly na následující dostupné stránce, zatímco třída <code>book</code> začíná na následující pravé stránce.

1.6.2. Balíky

Při psaní svého dokumentu můžete potřebovat věci, které základní L^AT_EX nevyřeší, např. vkládání grafiky, použití barevného textu nebo vysázení zdrojového kódu programu. V tom případě je potřeba schopnosti L^AT_EXu „zdokonalit“. Toto zdokonalování se děje pomocí balíků. Balíky se aktivují pomocí příkazu

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

kde *package* znamená jméno balíku a *options* je seznam klíčových slov, která zapínají speciální rysy balíku. Některé balíky jsou součástí základní distribuce L^AT_EX 2_ε (viz tabulka 1.3). Jiné se získávají zvlášť. Více informací o balících instalovaných na vašem počítači najdete ve své publikaci *Local Guide* [5]. Základním zdrojem informací o balících L^AT_EXu je *The L^AT_EX Companion* [3]. Zmíněná kniha obsahuje popisy stovek balíků, spolu s informací o tom, jak psát vlastní rozšíření L^AT_EX 2_ε.

Moderní T_EXové distribuce mají mnoho balíků předinstalovaných. Po nainstalování nejpoužívanější T_EXové distribuce T_EX Live na všechny klíčové operační systémy (Microsoft Windows, Unix, Mac OS X) můžete dokumentaci k balíkům získat příkazem `texdoc`.

1.6.3. Stránkové styly

L^AT_EX podporuje tři předdefinované kombinace záhlaví/paty stránek – tzv. stránkové styly. Parametr *style* příkazu

```
\pagestyle{style}
```

definuje, který z nich se má použít. Tabulka 1.4 ukazuje všechny tři předdefinované styly.

Styl aktuální stránky je možné změnit příkazem

```
\thispagestyle{style}
```

Popis toho, jak vytvořit vlastní záhlaví a paty najdete v *The L^AT_EX Companion* [3] a v sekci 4.4. na straně 85.

1.7. Soubory, na které můžete narazit

Při práci s L^AT_EXem se můžete setkat s řadou typů souborů (např. soubory, ve kterých je uložen kód balíků nebo pomocné soubory vygenerované L^AT_EXem při zpracování vašeho dokumentu). Jednotlivé typy souborů mají standardní přípony. Následující seznam stručně charakterizuje nejčastější typy souborů.

Tabulka 1.3: Některé z balíků distribuovaných s L^AT_EXem.

<code>doc</code>	Umožňuje dokumentovat L ^A T _E Xové programy. Je popsán v <code>doc.dtx</code> ^a a v <i>The L^AT_EX Companion</i> [3].
<code>exscale</code>	Poskytuje zvětšené verze rozšířených matematických fontů. Popsán v <code>ltxscale.dtx</code> .
<code>fontenc</code>	Specifikuje, které kódování fontů má L ^A T _E X použít. Popsán v <code>ltoutenc.dtx</code> .
<code>ifthen</code>	Poskytuje příkazy typu 'jestliže... pak udělej..., jinak udělej...' Popsán v <code>ifthen.dtx</code> a <i>The L^AT_EX Companion</i> [3].
<code>latexsym</code>	Zpřístupňuje L ^A T _E Xový font symbol. Popsán v <code>latexsym.dtx</code> a v <i>The L^AT_EX Companion</i> [3].
<code>makeidx</code>	Poskytuje příkazy produkující indexy. Popsán v sekci 4.3. a v <i>The L^AT_EX Companion</i> [3].
<code>syntonly</code>	Zpracuje dokument bez toho, aby produkoval výstup.
<code>inputenc</code>	Umožňuje specifikovat vstupní kódování, lze se setkat např. s kódováním ASCII, ISO Latin-1, ISO Latin-2, 437/850 IBM code pages, Apple Macintosh, Next, ANSI-Windows nebo uživatelsky definovaným. Balík je popsán v <code>inputenc.dtx</code> .

^aTento soubor by na vašem počítači měl být nainstalován a mělo by být možné získat z něj `dvi` soubor spuštěním `latex doc.dtx` v kterémkoliv adresáři, do něhož máte právo pro zápis. To samé platí pro všechny ostatní soubory zmíněné v této tabulce.

Tabulka 1.4: Předdefinované stránkové styly L^AT_EXu.

<code>plain</code>	Vypisuje čísla stránek doprostřed pat stránek (implicitní styl).
<code>headings</code>	Vypisuje název aktuální kapitoly a číslo stránky do záhlaví každé stránky. Pata stránky je prázdná; styl použitý v tomto dokumentu.
<code>empty</code>	Záhlaví i pata stránky jsou prázdné.

Seznam není úplný; pokud se vám zdá, že chybí důležitý typ, dejte mi prosím vědět.

- `.tex` vstupní soubor pro \LaTeX nebo \TeX . Lze ho přeložit pomocí `latex`, resp. `tex`, `pdflatex`, `xelatex`, `lualatex`...
- `.sty` Balík \LaTeX ových maker. Do \LaTeX ového dokumentu si ho nahrajete pomocí příkazu `\usepackage`.
- `.dtx` Dokumentovaný \TeX . Jedná se o hlavní distribuční formát pro stylové soubory \LaTeX u. Přeložíte-li soubor s příponou `.dtx`, obdržíte dokumentaci k makrům daného balíku (obsaženým v tomto `.dtx` souboru).
- `.ins` Instalátor pro soubory obsažené v příslušném `.dtx` souboru. Po stažení \LaTeX ového balíku ze sítě většinou získáte soubory s příponami `.dtx` a `.ins`. Spusťte \LaTeX na soubor `.ins` a získáte soubor s příponou `.dtx`.
- `.cls` Soubory s třídami definují, jak váš dokument vypadá. V \LaTeX ovém dokumentu příslušnou třídu vyberete příkazem `\documentclass`.
- `.fd` Soubor popisující font.

Následující soubory mohou být vygenerovány, spustíte-li \LaTeX na svém vstupním souboru:

- `.dvi` Device Independent File (Soubor nezávislý na zařízení). Jde o hlavní výstup kompilace pomocí \LaTeX u. Obsah souboru si můžete zobrazit pomocí prohlížeče DVI souborů nebo připravit pro tisk pomocí `dvips` nebo podobného programu.
- `.log` Obsahuje detailní popis věcí, ke kterým došlo během posledního překladu.
- `.toc` Obsahuje názvy všech oddílů dokumentu. Daný soubor je načten při příštím překladu a vysází se pomocí něj obsah.
- `.lof` Podobně jako `.toc` soubory, ale uchovává seznam obrázků.
- `.lot` Podobně jako `.toc` a `.lof` soubory, ale uchovává seznam tabulek.
- `.aux` Další soubor předávající informace z jednoho překladu do překladu dalšího. Kromě jiného jsou zde uloženy informace související s křížovými odkazy.
- `.idx` Obsahuje-li váš dokument rejstřík (index), \LaTeX uloží všechna slova, která byla v dokumentu označena pro uložení do indexu, do souboru s touto příponou. Tento soubor se zpracuje pomocí programu `makeindex` či `xindy`. Více informací o indexování lze najít v sekci 4.3. na straně 84.

`.ind` Výsledek zpracování souboru `.idx` programem `makeindex` či programem `xindy`. Tento soubor bude použit během další kompilace k vložení rejstříku do dokumentu.

`.ilg` Logovací soubor programu `makeindex`.

1.8. Velké projekty

Při psaní velkého dokumentu je někdy výhodné rozdělit si vstupní soubor na několik částí. To lze udělat pomocí dvou \LaTeX ových příkazů.

```
\include{filename}
```

Uvedením tohoto příkazu v těle dokumentu vložíte do aktuálního dokumentu obsah souboru `filename.tex`. Před vložením souboru \LaTeX začne novou stránku.

Druhý příkaz se používá v preambuli a říká \LaTeX u, aby vložil jen některé ze souborů uvedených v dokumentu u příkazu `\include`.

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

Po provedení tohoto příkazu budou vloženy jen ty ze souborů použitých jako argumenty příkazů `\include`, které jsou uvedeny jako parametry příkazu `\includeonly`. Mezi čárkami a jmény souborů v argumentu příkazu nesmí být žádné mezery.

Příkaz `\include` začne sazety vložený text na nové stránce. To je při použití `\includeonly` výhodné, protože stránkové zlomy se nezmění, ani když budou některé z vkládaných souborů vynechány. Pokud se vám ale toto chování nehodí, můžete použít příkaz

```
\input{filename}
```

Tento příkaz vloží zmíněný soubor bez přechodu na novou stránku.

Pro rychlou kontrolu svého dokumentu \LaTeX em můžete použít balík `syntonly`. \LaTeX projde vaším dokumentem a zkontroluje syntax a použití příkazů, nevytváří ale žádný (DVI či PDF) výstup. \LaTeX tímto způsobem zpracuje vstupní soubor rychleji. Použití balíku je následující:

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

Chcete-li, aby se výstup *generoval*, stačí pomocí procenta na začátku řádku zakomentovat druhou zmíněnou řádku.

Kapitola 2.

Sazba textu

Po přečtení předchozí kapitoly byste měli znát základní prvky dokumentu systému $\text{\LaTeX} 2_{\epsilon}$. V této kapitole doplním informace o zbytku reálně potřebných věcí.

2.1. Struktura textu a jazyka

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

Hlavním cílem psaní textu (vyjma některou DAAC¹ moderní literaturu) je předat myšlenky, informace nebo znalosti čtenáři. Text je srozumitelnější, pokud jsou myšlenky dobře strukturované a typografická forma odpovídá logické a významové struktuře obsahu.

\LaTeX se od jiných sázecích systémů liší v tom, že od uživatele vyžaduje popis logické a sémantické struktury textu. \LaTeX sám potom vytvoří typografickou reprezentaci textu podle *pravidel* uvedených v souboru s třídou dokumentu a různých stylových souborech.

Nejdůležitější *jednotka textu* v \LaTeX u (i v typografii) je odstavec. Odstavci se říká jednotka textu, protože je typografickou formou, která by měla vyjadřovat jednu soudržnou myšlenku nebo ideu. V následujících sekcích se dozvíte, jak vynutit řádkový zlom (např. pomocí `\`) a konec odstavce (např. pomocí prázdného řádku v zdrojovém souboru). S novou myšlenkou by měl začít i nový odstavec. Pokud jste ohledně dělení do odstavců na pochybách, představte si text jako přepravce ideí a myšlenek. Stejná myšlenka by neměla být rozdělena do několika odstavců, naopak objeví-li se v textu odstavce myšlenka nová, měla by být přesunuta do zvláštního odstavce.

Většina lidí podceňuje význam správného dělení textu do odstavců. Řada lidí dokonce neví, na základě čeho se toto dělení uskutečňuje nebo, zvláště v \LaTeX u, vytvářejí nové odstavce omylem. Tento poslední případ je jednoduché udělat hlavně tehdy, když jsou v textu použity rovnice. Podívejte

¹Different At All Cost (Odlišný za každou cenu), překlad zkratky UVA (Um's Verrecken Anders) ze švýcarské němčiny.

se na následující příklady a zkuste poznat, proč prázdné řádky (odstavcové zlomy) před a za rovnicí někdy uvedeny jsou a někdy ne. (Pokud těmto příkladům ještě úplně nerozumíte, přečtěte si nejdřív tuto a následující kapitolu a k této sekci se vraťte potom.)

```
% Example 1
\ldots když Einstein přišel s~vzorcem
\begin{equation}
  e = m \cdot c^2 \ ; \ ,
\end{equation}
což je nejznámější a~zároveň nejméně pochopený
matematický vzorec.
```

```
% Example 2
\ldots z~čehož plyne Kirchhoffův zákon o~uzlech:
\begin{equation}
  \sum_{k=1}^n I_k = 0 \ ; \ .
\end{equation}
```

Kirchhoffův zákon o~napětí lze odvodit\ldots

```
% Example 3
\ldots což má několik výhod.
```

```
\begin{equation}
  I_D = I_F - I_R
\end{equation}
je jádrem velmi odlišného tranzistorového modelu.
```

Menší textovou jednotkou je věta. V anglických textech bývá větší mezera za tečkou ukončující větu než za tečkou uvedenou za zkratkou. \LaTeX zkouší odhadnout, který z těchto dvou významů každá tečka má. Pokud je jeho odhad špatný, musíte udělat opravu. To je vysvětleno níže v této kapitole.

Strukturování textu se týká dokonce i částí vět. Většina jazyků má komplikovaná interpunkční pravidla, ale v mnoha jazycích (včetně němčiny a angličtiny) se můžete řídit jednoduchým vodítkem: krátká pauza v toku jazyka. Pokud si s čárkami nejste jistí, přečtěte si větu nahlas a krátce se zastavte na každé čárce. Pokud je někde pauza nepřirozená, vymažte příslušnou čárku. Cítíte-li, že byste se v některém místě měli nadechnout (nebo krátce zastavit), přidejte čárku.

Nakonec, odstavce textu by měly být logicky seskupeny do vyšších celků (kapitol, sekcí, podsekcí atd.). Typografický efekt napsání např. `\section{`

Struktura textu a jazyka} je zřejmý a je jasně vidět, jak se toto strukturování na vyšší úrovni používá.

2.2. Řádkový a stránkový zlom

2.2.1. Zarovnané odstavce

Knihy se často sází tak, že každý řádek má stejnou délku. Aby toho dosáhl, vkládá L^AT_EX nezbytné řádkové zlomy a mezery mezi slova a přitom optimalizuje s ohledem na obsah celého odstavce. Navíc dělí slova, která nelze dobře umístit na jednu řádku. To, jak jsou odstavce vysázeny, závisí na třídě dokumentu. Obyčejně se odsazuje první řádek odstavce a mezi dva odstavce se nekládá žádná dodatečná mezera. Více informací je uvedeno v sekci 6.3.2.

Ve speciálních případech musíme L^AT_EXu říct, aby rozdělil řádku:

```
\ nebo \newline
```

zahájí novou řádku (ale ne nový odstavec).

```
\*
```

navíc zakáže stránkový zlom za specifikovaným řádkovým zlomem.

```
\newpage
```

zahájí stránku.

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

indikuje místa, kde je možné provést řádkový, resp. stránkový zlom. Hodnotou nepovinného parametru je možné upřesnit, jak se má L^AT_EX zachovat. Hodnota může být mezi nula a čtyři. Hodnota čtyři znamená, že L^AT_EX nemusí zlom udělat, pokud by výsledek vypadal velmi špatně. Tyto *příkazy zlomu* ale nejsou to samé jako *příkazy new* (např. `\newline`). Použijete-li některý z „příkazů zlomu“, L^AT_EX se pokusí vyrovnat na řádku text před zlomem (jak je popsáno v následující sekci); nepříjemným výsledkem mohou být velké mezery na daném řádku. Pokud opravdu chcete začít novou řádku, resp. novou stránku, použijte příslušný „příkaz new“.

L^AT_EX se vždy snaží najít nejlepší možné řádkové zlomy. Pokud se mu nepodaří najít takové, které by splňovaly jeho vysoké standardy, nechá jednu z řádek odstavce přecházet přes pravý okraj a vypíše varování (`overfull hbox`). K tomu dochází hlavně tehdy, když se L^AT_EXu nepodaří najít vhodné

místo pro rozdělení slova.² Pomocí příkazu `\sloppy` je možné L^AT_EXu říct, aby trochu snížil své standardy. Dá se tím zabránit výše zmíněným přechívaným řádkám (pomocí zvětšení mezislovních mezer) a je zobrazeno varování (`underfull hbox`). Výsledek většinou nevypadá moc dobře. Příkazem `\fussy` vrátíme L^AT_EX zpět k jeho „vysokým standardům“.

2.2.2. Dělení slov

L^AT_EX se pokouší dělit slova vždy, když by to vedlo k „lépe vypadajícímu“ odstavci. Pokud v daném slově existují místa, kde lze slovo korektně rozdělit, ale L^AT_EXový algoritmus dělení slov tato místa nenajde, můžete pomocí následujících příkazů tato místa L^AT_EXu označit „ručně“.

Příkaz

```
\hyphenation{word list}
```

zařídí, že pro slova uvedená jako argument bude L^AT_EX uvažovat rozdělení jen v místech ve slově označených pomocí `-`. Argumenty příkazu by měla být slova obsahující jen normální písmena a znaky, které jsou L^AT_EXem za normální písmena považovány. Argument příkazu („tipy“ pro dělení daných slov) je uložen pro jazyk, který je aktivní ve chvíli, kdy je příkaz spuštěn. Umístíte-li tedy tento příkaz do preambule svého dokumentu, bude se vztahovat na angličtinu. Pokud příkaz umístíte za `\begin{document}` a používáte balík podporující sazbu v jiném jazyku (např. `babel`), pak se bude argument příkazu `\hyphenation` vztahovat na jazyk daným balíkem aktivovaný.

Následující příkaz zařídí, že bude možno dělit jak „hyphenation“, tak „Hyphenation“ a zakáže dělení slov „FORTRAN“, „Fortran“ a „fortran“. Argument příkazu `\hyphenation` nesmí obsahovat žádné speciální znaky nebo symboly.

Příklad:

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

Příkaz `\-` vloží na dané místo do daného slova „volitelné dělícítko“. Místa v daném slově takto označená budou jedinými místy, kde smí L^AT_EX dané slovo rozdělit. Tento způsob popisu výjimek pro dělení slov je obzvláště užitečný u slov, která obsahují speciální znaky (např. znaky s akcenty), protože L^AT_EX sám by takováto slova neumožnil dělit v žádném místě.

```
nej\-ne\-ob\-hos\-po\-da\-%
\v{r}o\-v\{a}\-va\-te1\-%
n\v{e}j\-\v{s}\{i}mu
```

```
nejneobhospodařovatelnějším
```

² Ačkoliv je vypsáno varování („Overfull hbox“) a vypsán problematický řádek, najít tento řádek v dokumentu nemusí být snadné. Použijete-li v příkazu `\documentclass` volbu `draft`, budou takového řádku označeny tlustou černou čarou v pravém okraji.

Chceme-li zabránit tomu, aby byla část textu rozdělena, můžeme tento text zadat jako argument příkazu

```
\mbox{text}
```

Brzy budu mít nové telefonní číslo. Bude to
`\mbox{0116 291 2319}`.

Parametr `\mbox{\emph{filename}}` by měl obsahovat jméno daného souboru.

```
Brzy budu mít nové telefonní číslo. Bude to 0116 291 2319.
```

```
Parametr filename by měl obsahovat jméno daného souboru.
```

Příkaz `\fbox` je podobný jako `\mbox`, ale navíc bude okolo obsahu nakreslen rámeček.

2.3. Předpřipravené řetězce

V některých z příkladů na předchozích stranách se objevily jednoduché \LaTeX ové příkazy pro sazbu speciálních řetězců:

Příkaz	Možný výsledek	Popis
<code>\today</code>	15. června 2011	Aktuální datum
<code>\TeX</code>	\TeX	Váš oblíbený sazeč
<code>\LaTeX</code>	\LaTeX	Jméno naší hry
<code>\LaTeXe</code>	$\text{\LaTeX} 2_{\epsilon}$	Její aktuální inkarnace

2.4. Zvláštní znaky a symboly

2.4.1. Uvozovky

Uvozovky byste *neměli* sázet pomocí znaku ". Pro otevírací a uzavírací uvozovku místo toho používáme v \LaTeX u speciální symboly: dva znaky ` (obrácený apostrof) pro otevírací uvozovku a dva znaky ' (apostrof) pro uzavírací uvozovku. Pro jednoduché uvozovky používáme jen jeden z této dvojice znaků.

```
‘‘Press the key ‘x’, please.’’
```

```
“Press the key ‘x’, please.”
```

Vysázeno výše vpravo to nevypadá nejlépe, ale opravdu jsme použili zpětný apostrof pro otevírací uvozovku a normální apostrof pro zavírací. Navzdory tomu, co zvolený font skutečně vysázel...

2.4.2. Pomlčky a spojovníky

Spojovník do L^AT_EXového dokumentu vložíme, uvedeme-li ve vstupním souboru jeden znak -. Uvedením dvou, resp. tří za sebou následujících znaků - vložíme pomlčky dvou různých velikostí. Uvedením znaku - uvnitř matematického vzorce vložíme do dokumentu matematický znak mínus:

```
je-li, anglicko-německý\\
na stranách 13--67\\
yes---or no? \\
$0$, $1$ a~$-1$
```

```
je-li, anglicko-německý
na stranách 13-67
yes-or no?
0, 1 a -1
```

Jména jednotlivých znaků jsou: spojovník (-), půlčtverčiková pomlčka (–), čtverčiková pomlčka (—) a matematický znak mínus (−).

2.4.3. Tilda (~)

Ve webových adresách se často objevuje znak tilda. V L^AT_EXu ho vygenerujete pomocí `\~`, ale výsledek `\~` asi není to, co chcete získat. Srovnajte:

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

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

2.4.4. Symbol stupně (°)

Tisk symbolu stupeň v „normálním“ L^AT_EXu.

```
Je-li opravdu
$-30\,^{\circ}\mathrm{C}$,
je asi zima\dots
```

```
Je-li opravdu -30 °C, je asi zima...
```

Balík `textcomp` generuje symbol stupně i pomocí příkazu `\textdegree` nebo – v kombinaci s C – pomocí `\textcelsius`.

```
30 \textcelsius{} je
86 \textdegree{}F.
```

```
30 °C je 86 °F.
```

2.4.5. Symbol Eura (€)

Mnoho současných fontů obsahuje symbol Eura. Po nahrání balíku `textcomp` v preambuli dokumentu

```
\usepackage{textcomp}
```

můžete tento symbol vysázet pomocí příkazu

```
\texteuro
```

Pokud váš font znak Eura neobsahuje, nebo pokud obsažen je, ale nelíbí se vám, můžete použít balík `eurosym`, který poskytuje oficiální podobu symbolu:

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

Pokud byste chtěli symbol Eura, který se hodí k vašemu fontu, použijte `gen` místo `official`.

Tabulka 2.1: Balík plný symbolů Eura.

LM+textcomp	<code>\texteuro</code>	€	€	€
eurosym	<code>\euro</code>	€	€	€
[gen]eurosym	<code>\euro</code>	€	€	€

2.4.6. Výpustky (...)

Na psacím stroji má znak čárky a tečky stejnou šířku jako každý jiný znak. U sázených textů jsou tyto znaky mnohem užší a sází se velmi blízko k předcházejícímu písmenu. Napsáním tří teček za sebou tedy výpustku nezískáme. Získáme ji ale pomocí příkazu:

```
\ldots
```

Takto ne ... ale takto ano:\\
New York, Tokyo,
Budapest, \ldots

Takto ne ... ale takto ano:
New York, Tokyo, Budapest, ...

2.4.7. Ligatury

Některé znaky, zapíšeme-li je ve vstupním souboru za sebou, budou vysázeny jako jediný speciální symbol (doporučujeme v ukázce využít zvětšení).

ff fi fl ffi... místo ff fi fl ffi ...

Jedná se o tzv. ligatury a spojení jednotlivých znaků ve vstupním souboru do speciálního symbolu (ligatury) dělá \LaTeX automaticky. Pokud si tyto speciální symboly nepřejeme, můžeme napsat `\mbox{}` mezi příslušné dva znaky. Potlačit ligatury můžeme chtít u slov skládajících se ze dvou slov.

```
\Large Not shelfful\
but shelf\mbox{ }ful
```

<p>Not shelfful but shelfful</p>

2.4.8. Akcenty a speciální znaky

\LaTeX podporuje používání akcentů a speciálních znaků mnoha jazyků. Tabulka 2.2 ukazuje všechny druhy akcentů aplikované na písmeno o. Akcenty jde samozřejmě stejně použít i na ostatní písmena.

Při přidávání akcentů nad písmena i a j je navíc potřeba odstranit tečky, které jsou součástí těchto písmen, aby s těmito akcenty nekolidovaly. Provede se to pomocí `\i` a `\j`.

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

<p>Hôtel, naïve, élève, smørrebrød, ¡Señorita!, Schönbrunner Schloß Straße</p>
--

2.5. Podpora pro neanglické jazyky

Když píšete dokumenty v jazycích jiných než je angličtina, je potřeba nakonfigurovat tři aspekty \LaTeX u:

1. Všechny automaticky generované textové řetězce³ je potřeba přizpůsobit danému jazyku. Pro mnoho jazyků tak lze učinit pomocí balíku `babel` od Johanna Braamse.
2. Je potřeba nastavit pravidla pro dělení slov v daném jazyce. Tento aspekt je trochu komplikovanější. Je potřeba znovu přeložit formátový soubor, který bude používat nové vzory dělení slov. Váš *Local Guide* [5] by měl obsahovat informace o tom, jak to provést.

³Nadpis obsahu, seznamu obrázků, ...

Tabulka 2.2: Akcenty a speciální znaky.

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

3. Je potřeba nastavit typografická pravidla specifická pro daný jazyk. Např. ve francouzštině je povinné psát mezeru před každý výskyt znaku dvojtečka (:).

Je-li váš systém nakonfigurován, aktivujte balík `babel` pomocí příkazu

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

za příkazem `\documentclass`. Na začátku každého překladu pomocí \LaTeX u pak bude zobrazen seznam *jazyků* zabudovaných do vašeho \LaTeX ového systému. Babel automaticky aktivuje příslušné vzory dělení slov pro jazyk, který si vyberete. Pokud váš \LaTeX ový formát nepodporuje dělení slov v jazyce, s kterým pracujete, `babel` bude fungovat i tak, ale dělení slov samozřejmě prováděno nebude, což mívá negativní vliv na vzhled vysázeného dokumentu.

Babel také pro některé jazyky poskytuje příkazy, které zjednodušují sazby speciálních znaků. Např. němčina obsahuje mnoho přehlásek (äöü). S nahreným bablem vložíte znak ö pomocí "o místo \".

Používáte-li `babel` s více jazyky, pak po provedení následujícího řádku vašeho dokumentu

```
\usepackage[jazykA,jazykB]{babel}
```

bude aktivní poslední z jazyků, uvedených jako argumenty `\usepackage` (tj. *jazykB*). Příkazem

```
\selectlanguage{languageA}
```

pak můžete změnit aktivní jazyk.

Většina moderních počítačů umožňuje zapsat písmena národních jazyků přímo z klávesnice. L^AT_EX na zacházení s vstupními kódováními různých skupin jazyků na různých platformách používá balík `inputenc`:

```
\usepackage[encoding]{inputenc}
```

Při používání tohoto kódování byste měli vzít v úvahu, že čtenáři vašeho dokumentu nemusí být schopni si správně zobrazit vaše vstupní soubory, protože používají jiné kódování. Např. německá přehláska ä je na OS/2 zakódována jako 132, na Unixových systémech používajících ISO-LATIN 2 jako 228, zatímco v kódování podle cyriliky cp1251 na Windows tento znak vůbec neexistuje. S `inputenc` byste proto měli zacházet obezřetně. Následující kódování mohou být užitečná (v závislosti na tom, který systém používáte).⁴

Operační systém	kódování	
	západní latinka	cyrilika
Mac	applemac	macukr
Unix	latin1	koi8-ru
Windows	ansinew	cp1251
DOS, OS/2	cp850	cp866nav

Obsahuje-li váš dokument texty ve více jazycích s nekompatibilními vstupními kódováními, můžete používat Unicode a balík `ucs`.

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

vám umožní vytvářet L^AT_EXové vstupní soubory v kódování `utf8x`, vícebytovém kódování kde každý znak je zakódován pomocí jednoho až čtyř bytů.

Jiným případem je kódování fontů, které definuje, na jakých pozicích v T_EXovém fontu jsou uloženy jednotlivé znaky. Více vstupních kódování může být namapováno do jednoho kódování fontu, což redukuje počet potřebných sad fontů. Kódování fontů se zařizuje pomocí balíku `fontenc`:

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

kde *encoding* je kódování fontu. Je možné nahrát více kódování.

Implicitním kódováním fontů v L^AT_EXu je OT1, což je kódování původních T_EXových fontů Computer Modern. Tyto fonty obsahují jen 128 znaků 7-bitové znakové sady ASCII. Když chceme používat akcentované znaky, můžeme je T_EXem vytvořit kombinací normálních znaků s akcenty. Výsledek sice vypadá dobře, ale slova s akcentovanými znaky nelze dělit. Navíc ne

⁴Více se o podporovaných vstupních kódováních pro latinkové a cyrilické jazyky můžete dočíst v dokumentaci k `inputenc.dtx` a k `cyinpenc.dtx`. Sekce 4.6. ukazuje, jak vyrobit dokumentaci k balíku.

všechny latinkové znaky lze vytvořit kombinací normálních znaků s akcenty, nemluvě o písmenech nelatinkových abeced, např. v řečtině nebo cyrilice.

Za účelem odstranění těchto nedostatků bylo navrženo několik sad osmibitových fontů odvozených z fontů Computer Modern. Fonty *Extended Cork* (EC) v kódování T1 obsahují písmena a interpunkční znaky většiny evropských jazyků založených na latince. Sada fontů LH obsahuje písmena potřebná k sazbě dokumentů v jazycích používajících cyriliku. Kvůli velkému počtu znaků cyriliky jsou tyto rozděleny do čtyř fontových kódování – T2A, T2B, T2C, a X2.⁵ Sada CB obsahuje fonty v kódování LGR a lze ji použít pro řecké texty.

Používáním těchto fontů můžete zlepšit, resp. umožnit dělení slov v neanglických dokumentech. Další výhodou používání fontů odvozených z CM je, že jsou dostupné ve všech váhách, tvarech a opticky škálované pro různé velikosti.

2.5.1. Podpora pro portugalštinu

autor: Demerson Andre Polli <polli@linux.ime.usp.br>

Aktivování dělení slov a změnu všech automatických textů do portugalštiny provedete příkazem:

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

Pokud jste v Brazílii, uveďte `brazilian` místo `portuguese`.

Protože v portugalštině existuje mnoho akcentů, může se vám hodit

```
\usepackage[latin1]{inputenc}
```

pomocí čehož je můžete snadno zadávat a také

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

pomocí kterého získáte korektní dělení slov.

V tabulce 2.3 je uveden text preambule, který je pro práci s portugalštinou potřeba použít. V příkladech používáme vstupní kódování `latin1`, takže nebudou fungovat na operačních systémech Mac nebo DOS. Stačí ale nastavit stejné kódování, jako má váš systém.

⁵Seznam jazyků podporovaných každým z těchto kódování lze najít v [11].

Tabulka 2.3: Preambule pro dokumenty v portugalštině.

```
\usepackage[portuguese]{babel}
\usepackage[latin1]{inputenc}
\usepackage[T1]{fontenc}
```

2.5.2. Podpora pro francouzštinu

autor: Daniel Flipo <daniel.flipo@univ-lille1.fr>

Několik rad pro ty, kteří pomocí L^AT_EXu vytvářejí dokumenty ve francouzštině: Nahrajte podporu francouzštiny pomocí příkazu:

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

Všimněte si, že z historických důvodů je jméno babelovského nastavení pro francouzštinu buď *frenchb* nebo *francais*, nikoliv *french*.

Tím jsme umožnili francouzské dělení slov (je-li váš L^AT_EXový systém řádně nakonfigurován). Také všechny automatické texty jsou převedeny do francouzštiny. `\chapter` vysází Chapitre, `\today` vysází aktuální datum ve francouzštině, atd. Je také aktivována sada nových příkazů, pomocí kterých můžete vstupní soubory ve francouzštině psát jednodušeji. Pro inspiraci se můžete podívat do tabulky 2.4:

Tabulka 2.4: Speciální příkazy pro francouzštinu.

<code>\og guillemets \fg{}</code>	<< guillemets >>
<code>M\up{me}, D\up{r}</code>	M ^{me} , D ^r
<code>1\ier{ }, 1\iere{ }, 1\ieres{ }</code>	1 ^{er} , 1 ^{re} , 1 ^{res}
<code>2\ieme{ } 4\iemes{ }</code>	2 ^e 4 ^{es}
<code>\No 1, \no 2</code>	N ^o 1, n ^o 2
<code>20~\degres C, 45\degres</code>	20 °C, 45°
<code>\bsc{M. ~Durand}</code>	M. DURAND
<code>\nombre{1234,56789}</code>	1234,56789

S přepnutím do francouzštiny se změní i vzhled seznamů. Více informací o tom, co volba `frenchb` v `babelu` dělá a jak lze přizpůsobit, spusťte L^AT_EX na soubor `frenchb.dtx` a přečtěte si vytvořený soubor `frenchb.dvi`.

2.5.3. Podpora pro němčinu

Několik tipů pro ty, kteří v \LaTeX u vytvářejí dokumenty v němčině: nahrajte podporu pro němčinu pomocí následujícího příkazu:

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

Tím je umožněno německé dělení slov (je-li váš \LaTeX ový systém nakonfigurován správně) a změněny automatické texty do němčiny. Z „Chapter“ se tak stává „Kapitel“. K dispozici je navíc sada nových příkazů, pomocí kterých můžete psát německé vstupní soubory rychleji dokonce i tehdy, když nepoužíváte balík `inputenc`. Pro inspiraci můžete nahlédnout do tabulky 2.5. Jestliže ale balík `inputenc` používáte, je možná lepší psát speciální německé symboly jednoduše vložením kódu daného znaku v daném kódování. (Nevýhodou tohoto postupu by snad mohly být komplikace při konverzi dokumentu do jiného kódování.)

Tabulka 2.5: Speciální německé znaky.

"a	ä	"s	ß
"‘	”	"’	“
"< nebo <code>\flqq</code>	«	"> nebo <code>\frqq</code>	»
<code>\flq</code>	<	<code>\frq</code>	>
<code>\dq</code>	”		

V německých knihách se často setkáte s francouzskými uvozovkami («guillemets»). Němečtí sazeči je ale používají jinak než francouzští. Citace v německé knize vypadá »takto«. V německé mluvící části Švýcarska ale sazeči používají «guillemets» stejným způsobem jako francouzští.

Vážným problémem vyskytujícím se při používání příkazů typu `\flq` je, že jestliže používáte font OT1 (což je implicitní font), guillemets vypadají jako matematický symbol “ \ll ”, což žádného sazeče nepotěší. Naproti tomu fonty kódované pomocí T1 obsahují příslušné symboly. Pokud tedy tento typ citací používáte, užíjte fontů v kódování T1, `\usepackage[T1]{fontenc}`.

2.5.4. Podpora pro korejštinu⁶

Abychom mohli sázet korejštinu, musíme vyřešit tři problémy:

⁶Nastínění řady problémů, s kterými se korejští uživatelé \LaTeX u musejí vypořádat. Za korejský tým překladatelů `lshort` tuto sekci sepsal Karnes KIM. Do angličtiny text přeložil SHIN Jungshik a zredukoval Tobi Oetiker.

1. Musíme být schopni vytvořit vstupní soubory v korejštině. Tyto soubory musí mít formát obyčejného textu, ale protože korejšтина používá vlastní znakovou sadu odlišnou od US-ASCII, budou při prohlížení pomocí normálního ASCII editoru tyto vstupní soubory vypadat „divoce“. Dvě nejčastěji používaná kódování korejštiny jsou EUC-KR a shora kompatibilní rozšíření používané na korejských MS-Windows, CP949/Windows-949/UHC. U těchto kódování kódy znaků US-ASCII reprezentují příslušné ASCII znaky, podobně jako u ostatních kompatibilních kódování jako např. ISO-8859-*x*, EUC-JP, Big5 nebo Shift_JIS. Naopak, slabiky Hangul, Hanjas (čínské znaky použité v korejštině), Hangul Jamos, Hiraganas, Katakanas, Greek a cyrilické znaky a další symboly a znaky odvozené z KS X 1001 jsou reprezentovány dvěma byty. První z nich má nastaven svůj nejvýznamnější bit. Do poloviny devadesátých let minulého století bylo velmi obtížné nastavit korejské národní prostředí na nelokalizovaných (nekorejských) operačních systémech. Chcete-li mít představu o tom, jak se tehdy korejšтина na nekorejských operačních systémech používala, podívejte se do dnes už velmi zastaralého dokumentu <http://jshin.net/faq>.
2. T_EX a L^AT_EX byly původně navrženy pro písmové systémy jejichž abecedy obsahovaly maximálně 256 znaků. Aby bylo možné sazety i jazyky s mnohem více znaky, jako např. korejštinu⁷ nebo čínštinu, byl vytvořen mechanismus subfontů. Ten dělí jeden ČJK font s tisíci nebo desetitisíci znaků na množinu subfontů po 256 značích. Pro korejštinu se široce používají tři balíky; H^AT_EX od UN Koaunghiho, h^AT_EXp od CHA Jaechoona a balík CJK od Wenera Lemberga.⁸ H^AT_EX

⁷ Korejský Hangul je systém založený na abecedě s čtrnácti základními souhláskami a deseti základními samohláskami (*jamos* [čamo]). Na rozdíl od latinky nebo cyriliky musí být jednotlivé znaky seřazeny do obdélníkových bloků velikých zhruba jako čínské znaky. Každý blok reprezentuje slabiku. Z konečného množství samohlásek a souhlásek je tedy možno vytvořit neomezeně mnoho slabik. Moderní korejské pravopisné standardy (pro Severní i Jižní Koreu) ale pro tvorbu uvádějí určitá omezení. Důsledkem je existence pouze konečného počtu korektních slabik. Korejské znakové kódování pro každou z těchto slabik definuje kódové pozice (KS X 1001:1998 and KS X 1002:1992). To znamená, že Hangul, i když je abecedního typu, se zpracovává jako čínšтина a japonšтина, s jejich desetitisíci obrázkovými/slovními znaky. ISO 10646/Unicode nabízí dvě cesty, jak Hangul použitý pro *moderní* korejštinu reprezentovat. První možností je pomocí zakódování spojených samohlásek Hangulu (abecedy: <http://www.unicode.org/charts/PDF/U1100.pdf>).

Druhou možností je zakódování všech korektních slabik Hangulu *moderní* korejštiny (<http://www.unicode.org/charts/PDF/UAC00.pdf>). Jednou z nejobávanějších výzev sazby korejštiny pomocí L^AT_EXu a příslušného sazecího systému je podpora středověké korejštiny – a případně budoucích korejštin – slabik, které lze reprezentovat jedině spojením *jamo* v Unicode. Lze doufat, že budoucí T_EXové systémy, např. Ω a Λ, přinesou řešení, takže i korejšti lingvisté a historici budou moct přejít z Microsoft Wordu, který má pro středověkou korejštinu slušnou podporu.

⁸ Je možné je získat z `language/korean/HLaTeX/`, `language/korean/CJK/` a <http://knot.kaist.ac.kr/htex/>

a $\text{H}\text{A}\text{T}\text{E}\text{X}$ p jsou specifické pro korejštinu a kromě podpory fontů poskytují i korejskou lokalizaci. Oba umějí zpracovat korejské vstupní soubory zakódované v EUC-KR. $\text{H}\text{A}\text{T}\text{E}\text{X}$, je-li použitý spolu se systémem Λ nebo Ω , umí zpracovat dokonce i vstupní soubory zakódované v CP949/Windows-949/UHC a UTF-8.

Balík CJK není specifický pro korejštinu. Umí zpracovat vstupní soubory v kódování UTF-8, EUC-KR a CP949/Windows-949/UHC a je možné s ním sázet vícejazyčné dokumenty (zvláště čínštinu, japonštinu a korejštinu). Na rozdíl od $\text{h}\text{A}\text{T}\text{E}\text{X}$ u neobsahuje balík CJK korejskou lokalizaci a nemá ani tolik speciálních korejských fontů.

Konečným cílem použití programů typu TEX a $\text{L}\text{A}\text{T}\text{E}\text{X}$ je vysázení dokumentů ‚esteticky‘ uspokojivým způsobem. Lze tvrdit, že nejdůležitější částí sazby je množina dobře navržených fontů. Distribuce $\text{H}\text{A}\text{T}\text{E}\text{X}$ u obsahuje fonty UHC POSTSCRIPT (deset rodin písem) a TrueType fonty Munhwabu⁹ (pět rodin písem). Balík CJK pracuje s množinou fontů používanou dřívější verzí $\text{H}\text{A}\text{T}\text{E}\text{X}$ u a může používat cyberbit TrueType font od Bitstreamu.

Abyste mohli své korejské texty vysázet pomocí balíku $\text{H}\text{A}\text{T}\text{E}\text{X}$, vložte následující řádek do preambule svého dokumentu:

```
\usepackage{hangul}
```

Tímto příkazem se zapne korejská lokalizace. Názvy kapitol, sekcí, podsekcí, obsah a tabulka obrázků budou všechny přeloženy do korejštiny a formátování dokumentu bude přizpůsobeno korejským konvencím. Navíc je aktivován automatický výběr „částic“. Korejštinu má dvojice částic-přípon, které jsou gramaticky ekvivalentní, ale liší se formou. To, kterou z nich použít závisí na tom, zda předcházející slabika končí samohláskou nebo souhláskou (zjednodušeně řečeno). Rodilí mluvčí nemají problém vybrat správnou částici, ale u odkazů a dalších automatických textů je tento výběr nemožný. Ruční upravování částic po každém přidání nebo odebrání odkazu nebo dalších úpravách je únavné. $\text{H}\text{A}\text{T}\text{E}\text{X}$ vás této zátěže zbaví.

Pokud chcete sázet korejštinu, ale korejskou lokalizaci nepotřebujete, stačí do preambule dokumentu vložit následující řádku:

```
\usepackage{hfont}
```

Více informací o sazbě korejštiny pomocí $\text{H}\text{A}\text{T}\text{E}\text{X}$ u lze najít v návodu k $\text{H}\text{A}\text{T}\text{E}\text{X}$ u. Dalším zdrojem informací jsou stránky korejské skupiny uživatelů TEX u (KTUG) na <http://www.ktug.or.kr/>. K dispozici je také korejský překlad tohoto manuálu.

⁹Korejské ministerstvo kultury.

2.5.5. Sazba řečtiny

autor: Nikolaos Pothitos <pothitos@di.uoa.gr>

Abyste mohli sázet řečtinu, musíte do preambule svého dokumentu vložit text uvedený v tabulce 2.6. Potom bude umožněno dělení řeckých slov a změněny automatické texty.¹⁰

Tabulka 2.6: Preambule pro řecké dokumenty.

```
\usepackage[english,greek]{babel}
\usepackage[iso-8859-7]{inputenc}
```

Výše zmíněný text v preambuli také zpřístupní množinu příkazů, pomocí které lze jednodušeji psát vstup v řečtině. Pro dočasné přepínání mezi řečtinou a angličtinou můžeme použít příkazy `\textlatin{anglický text}` a `\textgreek{řecký text}`, které oba vysázejí text předaný jako argument pomocí požadovaného kódování. Pro „dlouhodobé“ přepnutí použijte příkaz `\selectlanguage` popsany v předchozí sekci. Některé z řeckých interpunkčních znaků jsou uvedené v tabulce 2.7 Použijte `\euro` pro symbol Eura.

Tabulka 2.7: Greek Special Characters.

```
;    ·    ?    ;
((   <<   ))   >>
‘‘   ‘   ’’   ’
```

2.5.6. Podpora cyriliky

autor: Maksym Polyakov <polyama@myrealbox.com>

Verze 3.7h balíku `babel` obsahuje podporu kódování T2* a podporu sazby bulharských, ruských a ukrajinských textů pomocí písmen cyriliky.

¹⁰Použijete-li volbu `utf8x` balíku `inputenc`, L^AT_EX bude umět číst řecké a polytonické řecké znaky Unicode.

Podpora pro cyriliku je založena na mechanismech standardního L^AT_EXu a balíků fontenc a inputenc. Pokud ale chcete používat cyriliku v matematickém módu, budete potřebovat před balíkem fontenc nahrát balík mathtext.¹¹

```
\usepackage{mathtext}
\usepackage[T1,T2A]{fontenc}
\usepackage[koi8-ru]{inputenc}
\usepackage[english,bulgarian,russian,ukranian]{babel}
```

Balík babel obecně automaticky vybere implicitní kódování fontů, což pro výše zmíněné tři jazyky je kódování T2A. Dokumenty ale mohou obsahovat text ve více kódováních. Vícejazyčné dokumenty používající jak cyrilické tak latinkové jazyky je rozumné explicitně vložit v kódování latinkového fonu. Babel se postará o přepnutí do příslušného kódování fontů ve chvíli, kde je v dokumentu vybrán jiný jazyk.

Mimo zpřístupnění dělení slov, překladu automaticky generovaných textů a aktivování některých typografických pravidel týkajících se konkrétních jazyků (podobně jako příkaz `\frenchspacing` ve standardním L^AT_EXu) babel poskytuje příkazy umožňující sazbu podle standardů bulharštiny, ruštiny nebo ukrajinštiny.

K dispozici jsou interpunkční symboly pro všechny tři jazyky: cyrilická textová pomlčka (trochu užší než latinková a obklopená drobnými mezerami), pomlčka pro přímou řeč, uvozovky a příkazy usnadňující dělení slov, všechny jsou uvedeny v tabulce 2.8.

Ruské a ukrajinské volby babelu definují příkazy `\Asbuk` a `\asbuk`, které se chovají jako `\Alph` a `\alph`, ale vytvoří velká a malá písmena ruské nebo ukrajinské abecedy (v závislosti na tom, který z těchto jazyků je právě aktivní). Bulharská volba babelu zpřístupňuje příkazy `\enumBul` a `\enumLat` (`\enumEng`), které předefinovávají `\Alph` a `\alph` tak, že vytvoří písmena bulharské nebo latinské (anglické) abecedy. Implicitní výstup příkazů `\Alph` a `\alph` pro bulharštinu jsou znaky z bulharské abecedy.

2.5.7. Podpora pro mongolštinu

Pro sazbu mongolštiny pomocí L^AT_EXu máte dvě možnosti: buď vícejazyčný Babel, nebo MonT_EX Olivera Corffa.

MonT_EX podporuje jak cyrilici tak tradiční mongolské písmo. Abyste mohli použít příkazy MonT_EXu, přidejte:

```
\usepackage[language,encoding]{m1s}
```

do preamble. Pro generování popisů a dat v moderní mongolštině zadejte `xalx` jako volbu *language*. Pro psaní celého dokumentu v mongolštině je třeba

¹¹Používáte-li balíky $\mathcal{A}\mathcal{M}\mathcal{S}$ -L^AT_EX, nahraďte je také před fontenc a babelem.

Tabulka 2.8: Speciální definice pro bulharské, ruské a ukrajinské volby.

"	Zakaž ligaturu na tomto místě.
"-	Explicitní pomlčka, která nebrání dělení v daném slově.
"---	Pomlčka pro obyčejný text v cyrilice.
"--~	Pomlčka pro složená jména (příjmení) v cyrilice.
"--*	Pomlčka pro přímou řeč v cyrilice.
""	Podobné jako "-", ale nevytvoří žádný znak pomlčky (pro složená slova s pomlčkou, např. x-"y nebo některé jiné znaky jako „zakaž/povol“).
"~	Pro složené slovo bez možnosti zlomu.
"=	Pro složené slovo s možností zlomu, umožňující dělení v jednotlivých slovech.
" ,	Úzká mezera pro iniciály s místem zlomu v následujícím příjmení.
"‘	Pro německé levé dvojitě uvozovky (vypadají podobně jako: „).
"’	Pro německé pravé dvojitě uvozovky (vypadají takto: “).
"<	Pro francouzské levé dvojitě uvozovky (vypadají jako: <<).
">	Pro francouzské pravé dvojitě uvozovky (vypadají jako: >>).

jako *language* zadat hodnotu `bicig`. Volba jazyka `bicig` umožňuje vstupní metody „zjednodušené transliterace“.

Latinkový transliterační mód lze povolit a zakázat pomocí

```
\SetDocumentEncodingLMC
```

a

```
\SetDocumentEncodingNeutral
```

Na CTAN://tex-archive/language/mongolian/montex/doc získáte další informace o MonT_EXu.

Mongolské cyrilické písmo je podporováno i v `babelu`. Podporu pro mongolštinu aktivujete pomocí příkazů:

```
\usepackage[T2A]{fontenc}
\usepackage[mn]{inputenc}
\usepackage[mongolian]{babel}
```

kde `mn` je vstupní kódování `cp1251`. Pro modernější přístup místo toho použijte `utf8`.

2.6. Mezery mezi slovy

Aby mohly být řádky vysázených odstavců zarovnané (tj. mít stejnou šířku), \LaTeX mezi slova vkládá různě velké mezery. Na konci věty je obvykle vložena větší mezera, čímž se zlepši čitelnost. \LaTeX předpokládá, že věty končí buď tečkou, otazníkem nebo vykřičníkem. Pokud ale tečka následuje za velkým písmenem, \LaTeX předpokládá, že se o konec věty *nejedná* – ve většině případech totiž tato situace opravdu nastává u zkratk (uvedených uprostřed věty).

Chceme-li jiné chování než to popsané v předchozím odstavci, musíme se o ně sami postarat: Backslash před mezerou generuje mezeru, která se nikdy „nenatáhne“. Znak tilda ‘~’ vygeneruje mezeru, která také nemůže být roztažena a navíc v ní ani nemůže dojít k řádkovému zlomu. Příkaz `\@` uvedený před tečkou specifikuje, že tato tečka ukončuje větu, i když předchází velké písmeno.

```
Prof.~Smith byl šťastný, že
ji vidí\\
srv.~Obr.~5\\
Mám rád BASIC\@. Co vy?
```

Prof. Smith byl šťastný, že ji vidí srv. Obr. 5 Mám rád BASIC. Co vy?

Dodatečné mezery za tečkami lze zakázat pomocí příkazu

<code>\frenchspacing</code>

který \LaTeX u řekne, aby za tečku vložil *stejné množství* mezery bez ohledu na to, jestli se nachází na konci věty. To je běžné v neanglických jazycích (s výjimkou bibliografií). Použijete-li (jednou) `\frenchspacing`, příkaz `\@` už pak není nutné nikdy uvádět.

2.7. Titulky, kapitoly, sekce

Rozdělením dokumentu do kapitol, sekcí a podsekcí usnadníme čitateli orientaci v textu. \LaTeX pro tento účel nabízí několik příkazů, které jako svůj argument přijímají název dané kapitoly, sekce, ... Jen je nesmíte „pomíchat“.

Následující *oddílové* příkazy jsou dostupné ve třídě dokumentů `article`:

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

Chcete-li svůj dokument rozdělit do částí aniž byste ovlivnili číslování sekcí nebo kapitol, použijte

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

Používáte-li třídu `report` nebo `book`, máte na nejvyšší úrovni k dispozici ještě oddílový příkaz

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

Protože třída `article` neví nic o kapitolách (chapters), je jednoduché přidat články (articles) jako kapitoly do knihy (book). Mezery mezi sekcemi, číslování a velikost fontu nadpisů \LaTeX přizpůsobí automaticky.

Dva oddílové příkazy se trochu liší:

- Příkaz `\part` neovlivňuje sekvenci čísel kapitol.
- Příkaz `\appendix` nemá argumenty, pouze změní značení kapitol z čísel na písmena.¹²

Obsah je \LaTeX em vytvořen pomocí informací (jmén oddílů a čísel stran, kde jednotlivé oddíly začínají) shromážděných při předchozím překladu dokumentu. Příkaz

```
\tableofcontents
```

vytvoří obsah v místě, kde je zapsán. Aby čísla stránek uvedená v obsahu byla správná, je obvykle potřeba nový dokument \LaTeX em přeložit dvakrát. Někdy je dokonce potřeba dokument přeložit třikrát – \LaTeX vás na potřebu třetího přeložení případně sám upozorní.

Všechny oddílové příkazy zmíněné výše existují i ve „hvězdičkových“ verzích. Tyto verze zapíšeme tak, že za jméno příkazu uvedeme hvězdičku. Tyto verze se liší v tom, že nejsou uvedeny v obsahu a nejsou číslovány. Ukázka hvězdičkové verzi příkazu: `\section*{Help}`.

Normálně jsou názvy oddílů v obsahu zobrazeny stejně, jako jsou vysázeny na začátku daného oddílu. Příliš dlouhé názvy by se do obsahu nevešly

¹²Pro styl *article* se změní číslování sekcí.

a v takových případech je možné pomocí nepovinného argumentu zadat kratší text, který pro danou kapitolu bude v obsahu vysázen.

```
\chapter[Title for the table of contents]{A long
and especially boring title, shown in the text}
```

Titul celého dokumentu se generuje pomocí příkazu

```
\maketitle
```

Text popisující titul se vkládá pomocí příkazů

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

před zavoláním `\maketitle`. V příkazu `\author` může být seznam jmen oddělených příkazy `\and`. Na obrázku 1.2 najdete ukázkou použití některých z příkazů zmíněných výše.

Kromě právě zmíněných oddílových příkazů má \LaTeX 2 ϵ tři další příkazy ve třídě `book`. Jsou užitečné pro dělení publikací. Tyto příkazy mění názvy kapitol a číslování stránek tak, jak je obvyklé u knih:

`\frontmatter` by měl být uveden jako úplně první příkaz na začátku těla dokumentu (`\begin{document}`). Tento příkaz zapne číslování stránek římskými čísly a sekce nebudou číslovány, jako kdybyste používali hvězdičkované verze příkazů (např. `\chapter*{Preface}`). V obsahu ale sekce uvedeny *budou*.

`\mainmatter` se píše před první kapitolou knihy. Zapne číslování stránek arabskými čísly a stránky začne číslovat znovu od jedné.

`\appendix` značí místo, kde ve vašem dokumentu začíná přídatný materiál. Případné následující kapitoly budou značeny písmeny (a ne čísly).

`\backmatter` by mělo být uvedeno před posledními položkami vaší knihy (jako jsou seznam literatury a rejstřík). V standardních třídách dokumentů nemá tento příkaz žádný efekt.

2.8. Křížové odkazy

V knihách, zprávách a člancích se často odkazujeme (tj. používáme křížové odkazy) na obrázky, tabulky a speciální části textu. V \LaTeX u jsou k dispozici tyto příkazy pro křížové odkazy:

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

kde *marker* (též značka) je identifikátor vybraný uživatelem. \LaTeX nahradí `\ref` číslem sekce, podsekce, obrázku, tabulky nebo věty za kterými se vy-

skytuje příslušný příkaz `\label`. Příkaz `\pageref` vypíše číslo stránky, na které se vyskytl příkaz `\label`.¹³ Použijí se čísla vygenerovaná v předchozím překladu dokumentu, podobně jako u názvů oddílů.

Odkaz na tuto sekci
`\label{sec:this}` vypadá takto:
`\uv{viz sekce~\ref{sec:this}`
na straně~`\pageref{sec:this}.}`

Odkaz na tuto sekci vypadá takto: „viz
sekce 2.8. na straně 36.“

2.9. Poznámky pod čarou

Příkazem

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

vysázíme u paty stránky poznámku (tzv. poznámku pod čarou). Poznámky pod čarou vkládáme za¹⁴ slovo nebo větu, ke které se vztahují. Poznámky k větě nebo její části bychom tedy měli napsat za čárku nebo tečku.¹⁵

Poznámky pod čarou`\footnote{Toto`
je jedna z~nich.} jsou
u~lidí používajících
`\LaTeX{}` časté.

Poznámky pod čarou^a jsou u lidí používa-
jících \LaTeX časté.

^aToto je jedna z nich.

2.10. Zdůrazněná slova

Na psacím stroji důležitá slova zvýrazňujeme jejich podtržením.

```
\underline{text}
```

V knihách se ale slova zvýrazňují vysázením pomocí *italického* fontu. \LaTeX ke zvýraznění textu poskytuje příkaz

```
\emph{text}
```

Záleží na kontextu, co tento příkaz se svým argumentem udělá:

¹³Tyto příkazy neví, na co odkazují. `\label` jen uloží naposledy vygenerované číslo.

¹⁴Upozornění: *ne* před.

¹⁵Poznámky pod čarou odvádějí čtenářovu pozornost od hlavní části dokumentu – jsme nakonec zvědaví tvorové, proč tedy neříct všechno v hlavní části dokumentu?¹⁶

¹⁶Ukazatel směru nemusí sám jít tam, kam ukazuje. :-)

```
\emph{Budete-li
  zvýrazňovat uvnitř
  zvýrazňovaného textu,
  \LaTeX{} pro toto
  zvýraznění použije
  \emph{normální} font}
```

Budete-li zvýrazňovat uvnitř zvýrazňovaného textu, \LaTeX pro toto zvýraznění použije normální font

Všimněte si rozdílu mezi *zvýrazněním* a změnou *fontu*:

```
\textit{Máte možnost
  \emph{zvýraznit} text
  sázený pomocí italiky,}
\textsf{pomocí
  \emph{bezserifového} fontu,}
\texttt{nebo pomocí
  stylu \emph{psací stroj}.}
```

Máte možnost zvýraznit text sázený pomocí italiky, pomocí bezserifového fontu, nebo pomocí stylu psací stroj.

2.11. Prostředí

```
\begin{environment} text \end{environment}
```

Kde *environment* je jméno prostředí. Prostředí mohou být vnořená, ale nesmí být „pomíchaná“.

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

V následujících podsekcích vysvětlíme všechna důležitá prostředí.

2.11.1. Itemize, Enumerate a Description

Prostředí *itemize* je užitečné pro jednoduché seznamy (také se říká výčty), prostředí *enumerate* pro číslované seznamy a *description* pro popisy.

```

\flushleft
\begin{enumerate}
\item Seznamová prostředí můžete
míchat, jak se vám zlíbí:
\begin{itemize}
\item Ale výsledek nemusí
vypadat dobře.
\item[--] S pomlčkou.
\end{itemize}
\item Proto vezměte v úvahu:
\begin{description}
\item[Hloupé] věci
nezlepšíme tím, že je
dáme do seznamu.
\item[Chytré] věci ale
uvedením v seznamu mohou získat.
\end{description}
\end{enumerate}

```

1. Seznamová prostředí můžete míchat, jak se vám zlíbí:
 - Ale výsledek nemusí vypadat dobře.
 - S pomlčkou.
2. Proto vezměte v úvahu:

Hloupé věci nezlepšíme tím, že je dáme do seznamu.

Chytré věci ale uvedením v seznamu mohou získat.

2.11.2. Flushleft, Flushright a Center

Prostředí `flushleft` a `flushright` generují odstavce s řádky zarovnanými vlevo, resp. vpravo. Prostředí `center` generuje text s vycentrovanými řádky. Pokud nespecifikujete řádkový zlom sami, pomocí `\\`, `LATEX` řádky zalomí automaticky.

```

\begin{flushleft}
Tento text je\\ zarovnaný
doleva. \LaTeX{} se nesnaží
zajistit, aby každý řádek
byl stejně široký.
\end{flushleft}

```

Tento text je zarovnaný doleva. `LATEX` se nesnaží zajistit, aby každý řádek byl stejně široký.

```

\begin{flushright}
Tento text je
zarovnaný\\doprava.
\LaTeX{} se nesnaží zajistit,
aby každý řádek byl stejně
široký.
\end{flushright}

```

Tento text je zarovnaný doprava. `LATEX` se nesnaží zajistit, aby každý řádek byl stejně široký.

```

\begin{center}
V~centru\\země
\end{center}

```

V centru
země

2.11.3. Citace, citáty a verše

Prostředí `quote` je užitečné pro citace, důležité fráze a příklady.

Obecné pravidlo pro délku řádku je:
`\begin{quote}`
 V průměru by žádný řádek neměl být delší než 66 znaků.
`\end{quote}`
 Proto mají `\LaTeX`ové stránky implicitně tak široké okraje a proto se v novinách tiskne do sloupců.

Obecné pravidlo pro délku řádku je:

V průměru by žádný řádek neměl být delší než 66 znaků.

Proto mají `\LaTeX`ové stránky implicitně tak široké okraje a proto se v novinách tiskne do sloupců.

Existují dvě podobná prostředí: `quotation` a `verse`. Prostředí `quotation` je užitečné pro delší citace (přes několik odstavců), protože odsadí první řádku každého z těchto odstavců. Prostředí `verse` je užitečné pro básně (jejichž rozdělení do řádek se nedá dělat automaticky). Jejich řádky jsou odděleny pomocí `\\` na konci řádky a prázdné řádky oddělují sloky.

Nazpaměť znám pouze jednu anglickou báseň.
 Je o Humpty Dumptym.
`\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}`

Nazpaměť znám pouze jednu anglickou báseň. Je o Humpty Dumptym.

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. Abstrakty

Vědecké publikace obvykle začínají abstraktem – rychlým přehledem obsahu. `\LaTeX` pro tento účel poskytuje prostředí `abstract`, které obvykle používáme v dokumentech sázených pomocí třídy `article`.

`\begin{abstract}`
 Abstraktní abstrakt.
`\end{abstract}`

Abstraktní abstrakt.

2.11.5. Sazba doslovně

Text uzavřený mezi `\begin{verbatim}` a `\end{verbatim}` bude vysázen tak, jak je, jako kdyby byl napsán na psacím stroji. Všechny řádkové zlomy a mezery budou zachovány a jména L^AT_EXových příkazů budou vysázena, místo toho, aby se příslušné příkazy vykonaly.

Vprostřed normálního textu můžeme podobného chování docílit pomocí

```
\verb+text+
```

+ je jedním z možných oddělovačů. Můžete použít libovolný znak kromě písmen, * a mezery. Mnoho příkladů v této knize je sázeno pomocí tohoto příkazu.

Příkaz `\verb|\ldots| \ldots`

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

Příkaz `\ldots ...`

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

```
\begin{verbatim*}
hvězdičkováná verze
doslovného prostředí
zdůrazňuje mezery
v~textu
\end{verbatim*}
```

```
hvězdičkováná verze
doslovného prostředí
zdůrazňuje mezery
v~textu
```

Příkaz `\verb` lze s hvězdičkou použít také:

```
\verb*|like this :-)|
```

```
like this :-)
```

Prostředí `verbatim` a příkaz `\verb` někdy nejdou použít uvnitř parametru příkazu.

2.11.6. Tabular

Pomocí prostředí `tabular` lze sázet nádherné tabulky s případnými horizontálními a vertikálními linkami. Šířku sloupců L^AT_EX určí automaticky.

Argument *table spec* příkazu

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

definuje formát tabulky. Sloupec tabulky zarovnaný vlevo se specifikuje pomocí `l`, sloupec zarovnaný vpravo pomocí `r` a vycentrovaný sloupec po-

mocí `c`; sloupec obsahující zarovnaný text (výsledek automatického řádkového zlomu L^AT_EXu) pomocí `p{width}` a vertikální linka pomocí `|`.

Je-li text pro daný sloupec příliš široký, L^AT_EX ho *nebude* automaticky lámat. Pro přizpůsobení textu šířce sloupce je potřeba použít `p{width}`, kde se pak s textem bude zacházet stejně, jako v normálním odstavci.

Argument *pos* specifikuje vertikální pozici tabulky vzhledem k účarí okolního textu. Můžete použít některý ze znaků `t`, `b` a `c` (pro zarovnání horní části, resp. spodní části, resp. středu tabulky).

Uvnitř prostředí `tabular` znamená znak `&` oddělovač sloupců, `\` začíná novou řádku a `\hline` vloží horizontální linku. Neúplné linky můžeme vložit pomocí `\cline{i-j}`, kde *i* a *j* jsou čísla sloupců, mezi kterými by linka měla být natažena. Při užití `czech` v `babelu` vypneme navíc lokálně či globálně aktivní divis: `\shorthandoff{-}`.

```
\begin{tabular}{|r|l|}
\hline
7C0 & šestnáctkově \\
3700 & osmičkově \\
11111000000 & binárně \\
\hline \hline
1984 & desítkově \\
\hline
\end{tabular}
```

7C0	šestnáctkově
3700	osmičkově
11111000000	binárně
1984	desítkově

```
\begin{tabular}{|p{4.7cm}|}
\hline
Menší demonstrace odstavce
uvnitř tabulky. Výsledek snad
bude vypadat přijatelně. \\
\hline
\end{tabular}
```

Menší demonstrace odstavce uvnitř tabulky. Výsledek snad bude vypadat přijatelně.

Oddělovač sloupců lze specifikovat pomocí konstrukce `@{...}`. Tento příkaz potlačí veškeré mezisloupcové mezery a nahradí je materiálem uvedeným mezi složenými závorkami. Jedno běžné použití tohoto příkazu je vysvětleno níže (zarovnání desetinných čísel). Jinou možnou aplikací je potlačení úvodních mezer v tabulce pomocí `@{}`.

```
\begin{tabular}{@{} l @{}}
\hline
žádné úvodní mezery \\
\hline
\end{tabular}
```

žádné úvodní mezery

```
\begin{tabular}{l}
\hline
úvodní a~závěrečné mezery\\
\hline
\end{tabular}
```

úvodní a závěrečné mezery

V L^AT_EXu není vestavěna podpora pro zarovnávání desetinných čísel¹⁷, jednou z možností, jak čísla zarovnat je použít tabulku se dvěma sloupci: vpravo zarovnaná celá část desetinného čísla a vlevo zarovnaná příslušná desetinná část. Příkaz `@{,}` ve specifikaci sloupců `\begin{tabular}` nahradí normální mezisloupcové mezery čárkou a tabulka bude vypadat jako jeden sloupec čísel zarovnaných na desetinnou čárku. Nezapomeňte oddělit celou a desetinnou část jednotlivých čísel místo desetinou čárkou oddělovačem sloupců (`&`)! Název sloupce umístíme nad náš číselný „sloupec“ pomocí příkazu `\multicolumn`. Více o možnostech tabulkové sazby balíček `tabu`.

```
\begin{tabular}{c r @{,} l}
výraz obsahující Pi &
\multicolumn{2}{c}{Hodnota} \\
\hline
$\pi$ & 3&1416 & \\
$\pi^{\pi}$ & 36&46 & \\
$(\pi^{\pi})^{\pi}$ & 80662&7 & \\
\end{tabular}
```

výraz obsahující Pi	Hodnota
π	3,1416
π^{π}	36,46
$(\pi^{\pi})^{\pi}$	80662,7

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

Ene	
Mene	Muh!

Materiál vysázený pomocí prostředí `tabular` vždy zůstane pohromadě na jedné stránce. Chcete-li sázet vícestránkové tabulky, je třeba použít prostředí `longtable`.

L^AT_EXové tabulky mohou někdy vypadat trochu „zhuštěně“. „Rozvolnění“ lze provést nastavením vyšších hodnot `\arraystretch` a `\tabcolsep`.

¹⁷Pokud ale máte nainstalovanou kolekci `tools`, podívejte se na balík `dcolumn`.


```

\begin{tabular}{|l|}
\hline
Tyto řádky\\\hline
jsou namačkané\\\hline
\end{tabular}

{\renewcommand{\arraystretch}
{1.5}
\renewcommand{\tabcolsep}{0.2cm}
\begin{tabular}{|l|}
\hline
méně namačkané\\\hline
rozložení tabulky\\\hline
\end{tabular}}

```

Tyto řádky
jsou namačkané
méně namačkané
rozložení tabulky

Pokud potřebujete zvětšit výšku jen jedné řádky tabulky, přidejte neviditelnou vertikální podpěru¹⁸ pomocí linky s nulovou šířkou.

```

\begin{tabular}{|c|}
\hline
\hline
\rule{1pt}{4ex}Pitprop \ldots\
\hline
\rule{0pt}{4ex}Strut\
\hline
\end{tabular}

```

Pitprop ...
Strut

2.12. Plovoucí objekty

Dnešní publikace často obsahují hodně obrázků a tabulek. S těmi se musí zacházet zvláštním způsobem, protože na rozdíl od normálního textu nemohou být vždy rozděleny na více stránek. Jednoduché ale neuspokojivé řešení je začít novou stránku vždy, když je potřeba vysázet obrázek nebo tabulku, které se nevezou na aktuální stránku. Problém tohoto přístupu je, že části stránek pak zůstávají prázdné, což špatně vypadá.

Uspokojivým řešením je nechat obrázky a tabulky, které se nevezou na aktuální stránku, „doplavat“ na některou z následujících stránek, přičemž aktuální stránka se zaplní textem, který ve vstupním souboru následuje za daným obrázkem nebo tabulkou. \LaTeX pro *plovoucí objekty* nabízí dvě prostředí – jedno pro tabulky a jedno pro obrázky. Abychom mohli plně využít možností těchto dvou prostředí, je důležité, abychom zhruba věděli, jak \LaTeX s plovoucími objekty vnitřně zachází. Jinak bychom měli problémy odhadnout, kam \LaTeX konkrétní plovoucí objekty umístí.

¹⁸V profesionální sazbě se jí často říká strut.

Začneme náš výklad s \LaTeX ovými příkazy, které jsou pro plovoucí objekty k dispozici.

S jakýmkoliv materiálem uvedeným v prostředí `figure` nebo `table` se bude zacházet jako s plovoucím objektem. Obě prostředí pro plovoucí objekty poskytují nepovinný parametr

```
\begin{figure}[specifikace umístění] or \begin{table}[...]
```

nazvaný *specifikace umístění*. Tímto parametrem můžeme \LaTeX u říct o místech, do kterých smí daný plovoucí objekt nechat doplavat. *Specifikace umístění* se skládá z řetězce *povolенок pro umístění plovoucího objektu*. Viz tabulka 2.9.

Tabulku můžeme začít např. následujícím řádkem:

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

Povolenka umístění `![hbp]` říká \LaTeX u, že může tabulku umístit buď přímo v aktuálním místě (`h`), nebo naspodu nějaké strany (`b`), nebo na zvláštní stránce obsahující jen plovoucí objekty (`p`) – to vše i tehdy, když výsledek nebude vypadat nejlépe (viz `!`). Neuvedeme-li žádnou povolenku, \LaTeX se chová, jako bychom uvedli `[tbp]`.

\LaTeX umístí každý plovoucí objekt, na který při zpracování dokumentu narazí, podle povolenky umístění zadané autorem. Nejde-li objekt přidat do aktuálního místa na stránce, je přidán buď do fronty *obrázků* nebo do

Tabulka 2.9: Povolenky pro umístění plovoucího objektu.

Pozice	Povolení umístit plovoucí objekt...
<code>h</code>	<i>Zde</i> (here) přesně na místo, kde byl daný text uveden ve vstupním souboru. Toto je užitečné zejména při použití malých fontů.
<code>t</code>	Na <i>vrcholu</i> (top) strany.
<code>b</code>	Na <i>spodu</i> (bottom) strany.
<code>p</code>	Na zvláštní <i>stránce</i> (page), která bude obsahovat jen plovoucí objekty.
<code>!</code>	Bez braní v úvahu většiny vnitřních parametrů ^a , které by mohly zabránit v umístění tohoto plovoucího objektu.

Poznámka: `pt` a `em` jsou \TeX ové jednotky. Více informací je uvedeno v tabulce 6.5 na straně 125.

^aNapř. maximální počet plovoucích objektů, který může být umístěn na jednu stranu.

fronty *tabulek*¹⁹. Když zahájí novou stránku, L^AT_EX nejdřív zkontroluje, jestli je v některé frontě „na řadě“ objekt, který má povoleno být umístěn na zvláštní stránce (která obsahuje jen plovoucí objekty). Jestliže takový objekt přítomen není, první objekt v každé frontě je uvažován jako kdyby se právě objevil ve vstupním souboru: L^AT_EX znovu zkouší umístit tento objekt podle příslušných povolenek (kromě povolenky ‚h‘, která se už u objektu ve frontě vyskytnout nemůže). Jakýkoliv nový plovoucí objekt, který se objeví v textu, je umístěn do příslušné fronty. L^AT_EX přidává plovoucí objekty do výstupu přesně v tom pořadí, v kterém na ně ve vstupním souboru narazí. Z toho vyplývá, že obrázek, který se nepodaří umístit na aktuální stranu, „zablokuje“ i ostatní obrázky (protože ty nemohou být umístěny dřív, než obrázek, který je ve frontě před nimi).

Proto:

Jestliže L^AT_EX neumístí plovoucí objekt tak, jak jste očekávali, často je to proto, že jeden z plovoucích objektů zablokoval jednu z front plovoucích objektů.

L^AT_EX sice umožňuje zadat povolenku umístění o jedné položce, ale ta způsobuje problémy. Jestliže plovoucí objekt nelze pomocí dané položky umístit, jsou on i všechny následující zablokovány. Obzvlášť *nevhodná* je položka [h] – je to tak špatná možnost, že v novějších verzích L^AT_EXu je dokonce automaticky změněna na [ht].

Tím jsme probrali to obtížné. Nyní se zmíníme o několika věcech souvisejících s prostředím `table` a `figure`. Příkazem

```
\caption{caption text}
```

můžeme definovat titulek daného plovoucího objektu. L^AT_EX připojí pořadové číslo a řetězec „Obrázek“ nebo „Tabulka“.

Příkazy

```
\listoffigures a \listoftables
```

pracují podobně jako příkaz `\tableofcontents` – vysází seznam obrázků, resp. tabulek. Tyto seznamy zobrazí celý text uvedený jako titulek jednotlivých objektů. Pokud tedy používáte dlouhé titulky, měli byste zadat i jejich kratší verze, Krátkou verzi můžete uvést do hranatých závorek za příkazem `\caption`.

```
\caption[Short]{LLLLLoooooonnnnngggg}
```

¹⁹Fronty jsou klasické FIFO struktury – „Kdo první vejde, první odejde.“

Daný plovoucí objekt „identifikujete“ pomocí příkazu `\label` a můžete se pak na něj ve svém dokumentu odkazovat pomocí příkazu `\ref`. Příkaz `\label` musí být uveden *za* příkazem `\caption`, aby se vztahoval na daný plovoucí objekt.

Následující příklad nakreslí čtverec a vloží ho do dokumentu. Takto můžete v dokumentu rezervovat místo pro obrázky, které budete později do hotového dokumentu vkládat.

```
Obrázek~\ref{white} je příkladem pop artu.
\begin{figure}[!hbt]
\makebox[\textwidth]{\framebox[5cm]{\rule{0pt}{5cm}}}
\caption{Pět krát pět centimetrů.\label{white}}
\end{figure}
```

V předchozím příkladu \LaTeX udělá *všechno možné* (viz !) pro to, aby obrázek umístil *v aktuálním místě* (h).²⁰ Pokud se to ani tak nepodaří, \LaTeX zkusí obrázek umístit *naspod* (b) strany. Pokud se nepodaří ani to, \LaTeX zkusí obrázek umístit *nahoru* (t) na aktuální straně. Nepodaří-li se ani to, \LaTeX zjistí, jestli je možné vytvořit „plovoucí stránku“ obsahující tento obrázek a případně některé z tabulek ve frontě tabulek. Není-li pro speciální stránku obsahující jen plovoucí objekty dost těchto objektů k dispozici, \LaTeX zahájí novou stranu a opět zachází s obrázkem stejným způsobem, jako kdyby se právě objevil v textu.

Někdy je nutné použít příkaz

`\clearpage` nebo dokonce `\cleardoublepage`

Tento příkaz říká \LaTeX u, aby hned umístil všechny plovoucí objekty přítomné ve frontách a potom začal novou stránku. Příkaz `\cleardoublepage` udělá to samé, ale navíc přejde na novou pravou stranu (nejbližší lichou).

Později se v tomto dokumentu dočtete o tom, jak do svého dokumentu $\text{\LaTeX} 2_{\epsilon}$ vložit kresby ve formátu POSTSCRIPT.

2.13. Chránění „zranitelných“ příkazů

Text, který zadáme jako argument příkazů typu `\caption` nebo `\section`, se může v dokumentu objevit na více místech (např. v obsahu i v těle dokumentu). Pokud některé příkazy uvedeme jako argumenty příkazů typu `\section`, zpracování našeho dokumentu selže. Těmto příkazům, které jako parametry selžou, říkáme zranitelné příkazy. Patří mezi ně např. `\footnote` nebo `\phantom`. Pokud ale zranitelné příkazy „ochráníme“ tím, že před nimi uvedeme příkaz `\protect`, jako argumenty příkazů typu `\section` je už použít můžeme.

²⁰Za předpokladu, že fronta obrázků je prázdná.

Příkaz `\protect` se vztahuje jen na příkaz, který za ním bezprostředně následuje, ale ne na jeho argumenty. Ve většině případu neuškodí, když `\protect` uvedeme navíc.

```
\section{I am considerate  
  \protect\footnote{and protect my footnotes}}
```


Kapitola 3.

Sazba matematických vzorců

Jsme už připraveni, abychom v této kapitole mohli předvést hlavní sílu TEX : matematickou sazbu. Ale upozorňuji, že nepůjdeme příliš do hloubky. Věci vysvětlené v této kapitole vám sice většinou budou stačit, ale je možné, že nevyřeší některé konkrétní věci, které byste s matematickou sazbou potřebovali udělat. Je ale velmi pravděpodobné, že váš problém řeší $\mathcal{A}\mathcal{M}\mathcal{S}\text{-L}\text{A}\text{T}\text{E}\text{X}$.

3.1. Kolekce $\mathcal{A}\mathcal{M}\mathcal{S}\text{-L}\text{A}\text{T}\text{E}\text{X}$

Chcete-li sázet (pokročilou) matematiku, měli byste použít $\mathcal{A}\mathcal{M}\mathcal{S}\text{-L}\text{A}\text{T}\text{E}\text{X}$. $\mathcal{A}\mathcal{M}\mathcal{S}\text{-L}\text{A}\text{T}\text{E}\text{X}$ je kolekci balíků a tříd pro matematickou sazbu, z nichž my zde většinou budeme používat balík `amsmath`. $\mathcal{A}\mathcal{M}\mathcal{S}\text{-L}\text{A}\text{T}\text{E}\text{X}$ je spravován *Americkou matematickou společností* a je široce používán pro sazbu matematiky. Samotný $\text{L}\text{A}\text{T}\text{E}\text{X}$ sice poskytuje některé základní rysy a prostředí pro sazbu matematiky, ale jsou omezené (nebo naopak: $\mathcal{A}\mathcal{M}\mathcal{S}\text{-L}\text{A}\text{T}\text{E}\text{X}$ je *neomezený*) a v některých případech nekonzistentní.

$\mathcal{A}\mathcal{M}\mathcal{S}\text{-L}\text{A}\text{T}\text{E}\text{X}$ je součástí vyžadované distribuce a skutečně je obsažen ve všech současných $\text{L}\text{A}\text{T}\text{E}\text{X}$ ových distribucích.¹ V této kapitole budeme vždy předpokládat, že `amsmath` byl nahrán v preambuli vašeho dokumentu (pomocí `\usepackage{amsmath}`).

3.2. Jednoduché rovnice

Jsou dvě možnosti, jak sázet matematické vzorce: v rámci textu odstavce (*styl text*), nebo na samostatném řádku (*styl display*). Vzorce v *textovém stylu* uvádíme mezi znaky `$` a `$`:

¹Pokud ve vaší distribuci chybí, obraťte se na CTAN://tex-archive/macros/latex/required/amslatex.

Sečti a na druhou
 a^2 na druhou
 a^2 získej tak c na druhou.
 Nebo, matematicky řečeno:
 $a^2 + b^2 = c^2$.

Sečti a na druhou a b na druhou a získej tak c na druhou. Nebo, matematicky řečeno: $a^2 + b^2 = c^2$.

`\TeX` se vyslovuje jako
 $\tau\varepsilon\chi$.
 100m^3 vody.
 Toto vychází z mého \heartsuit .

`\TeX` se vyslovuje jako $\tau\varepsilon\chi$.
 100m^3 vody.
 Toto vychází z mého \heartsuit .

Chcete-li, aby vaše větší rovnice byly vysázeny odděleně od zbytku odstavce, je lepší zobrazit je ve stylu `display` spíš než „rozsekat“ kvůli nim odstavec. Toho dosáhnete tím, že tyto rovnice uvedete mezi `\begin{equation}` a `\end{equation}`.² Nyní si můžete označit (`\label`) číslo rovnice a odkázat se na ně v jiném místě textu pomocí příkazu `\eqref`. Chcete-li rovnici pojmenovat explicitně, můžete místo `\label` použít `\tag` (`\tag` ale nemůžete kombinovat s `\eqref`).

Sečti a na druhou
 a^2 na druhou
 a^2 získej tak c na druhou.
 Nebo, matematicky řečeno:
`\begin{equation}`
 $a^2 + b^2 = c^2$
`\end{equation}`
 Einstein říká, že
`\begin{equation}`
 $E = mc^2$ `\label{clever}`
`\end{equation}`
 Nikdy ale neřekl, že
`\begin{equation}`
 $1 + 1 = 3$ `\tag{lež}`
`\end{equation}`
 Toto je odkaz na
`\eqref{clever}`.

Sečti a na druhou a b na druhou a získej tak c na druhou. Nebo, matematicky řečeno:

$$a^2 + b^2 = c^2 \quad (3.1)$$

 Einstein říká, že

$$E = mc^2 \quad (3.2)$$

 Nikdy ale neřekl, že

$$1 + 1 = 3 \quad (\text{lež})$$

 Toto je odkaz na (3.2).

Nechete-li, aby \LaTeX rovnice čísloval, použijte „hvězdičkovanou“ verzi příkazu, `equation*`, nebo ještě jednodušeji, uveďte rovnici mezi `\[a \]`:³

² Jedná se o příkaz definovaný v `amsmath`. Pokud byste náhodou k tomuto balíku neměli přístup, můžete místo něj použít \LaTeX ové prostředí `displaymath`.

³ Opět se jedná o definice z `amsmath`. Pokud tento balík nemáte k dispozici, použijte místo něj \LaTeX ové prostředí `equation`. Stejná jména příkazů objevující se jak v `amsmath` tak v \LaTeX u trochu matou, ale zase takový problém to není, protože každý stejně používá `amsmath`. Obvykle je nejlepší nahrát tento balík hned na začátku dokumentu, protože když se rozhodnete přidat ho až později, může \LaTeX ová sazba nečíslovaných rovnic kolidovat s \mathcal{AMS} - \LaTeX ovou sazbou číslovaných rovnic.

Sečti a na druhou
 $a^2 + b^2$ na druhou
 $a^2 + b^2$ získkej tak c na druhou.
 Nebo, matematicky řečeno:

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

 nebo to samé stručněji:

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

Sečti a na druhou a b na druhou a získkej tak c na druhou. Nebo, matematicky řečeno:

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

nebo to samé stručněji:

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

Pokud ale váš dokument obsahuje mnoho rovnic a budete je všechny sázet pomocí $\left[a \right]$, L^AT_EXový zdrojový kód vašeho dokumentu bude nepřehledný. Proto v takovém případě doporučujeme používat `equation`, resp. `equation*`.

Všimněte si, jak se liší styl sazby rovnic v textovém stylu a ve stylu `display`:

Toto je textový styl:

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

A toto je styl `display`:

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

Toto je textový styl: $\lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6}$. A toto je styl `display`:

$$\lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6} \quad (3.3)$$

Vysoké nebo hluboké matematické vzorce v textovém stylu uzavírejte do `\smash`. L^AT_EX pak bude ignorovat výšku a hloubku těchto výrazů a rovnoměrné řádkování dokumentu zůstane zachováno.

Matematický vzorec $d_{o_{1_u}}$ následovaný vzorcem $n^{a^{h^{o^r u}}}$. Naproti tomu `\uv{smashed}` výraz `\smash{ $d_{o_{1_u}}$ }` následovaný výrazem `\smash{ $n^{a^{h^{o^r u}}}$ }`.

Matematický vzorec $d_{o_{1_u}}$ následovaný vzorcem $n^{a^{h^{o^r u}}}$. Naproti tomu „smashed“ výraz $d_{o_{1_u}}$ následovaný výrazem $n^{a^{h^{o^r u}}}$.

3.2.1. Matematický mód

Rozdíly jsou i mezi *matematickým módem* a *textovým módem*. V *matematickém módu* např.:

1. Většina mezer a řádkových zlomů nemá žádný efekt, protože všechny mezery jsou odvozeny z matematických výrazů nebo je třeba je uvést

pomocí speciálních příkazů, např. `\,`, `\quad` nebo `\qquad` (k tomuto se ještě vrátíme, v sekci 3.7.).

2. Nejsou povoleny prázdné řádky. Na vzorec připadá jen jeden odstavec.
3. Každé písmeno je považováno za jméno proměnné a jako takové je vysázeno. Chcete-li uvnitř vzorce vysázet normální text (normální vzpřímený font a normální mezerování), musíte ho uvést pomocí příkazu `\text{...}`, viz také sekci 3.8. na straně 67.

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

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

```
x^2 \geq 0 \quad \text{for all } x \in \mathbf{R}
```

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

Matematici jsou někdy velmi úzkostliví ohledně toho, které symboly jsou použity: podle konvencí by bylo správné použít font Blackboard Bold, pomocí `\mathbb` z balíku `amssymb`.⁴ Předchozí příklad tedy můžeme vylepšit do následující formy:

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

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

Více matematických fontů je uvedeno v tabulce 3.14 na straně 76 a v tabulce 6.4 na straně 121.

3.3. Stavební bloky matematických vzorců

V této sekci popíšeme nejdůležitější příkazy použité v matematické sazbě. Většina z nich nevyžaduje `amsmath` (na výjimky upozorníme), ale nahrajte tento balík do svého dokumentu i tak.

Malá řecká písmena se vkládají jako `\alpha`, `\beta`, `\gamma`, ..., velká písmena pak jako `\Gamma`, `\Delta`, ...⁵

Seznam řeckých písmen je uveden v tabulce 3.2 na straně 72.

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

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

⁴Balíček `amssymb` není součástí kolekce `AMS-LATEX`, ale přesto je pravděpodobné, že je přítomen ve vaší `LATEX`ové distribuci. Pokud přítomen není, můžete ho získat na `CTAN:/fonts/amsmath/latex/`.

⁵Speciální vkládání velkého Alpha, Beta a jiných není v `LATEX 2ε` definováno, protože tato písmena vypadají stejně jako normální latinkové A, B, atd. Po zavedení nového matematického kódování zde dojde ke změně.


```


$$\underbrace{\overbrace{a+b+c}^6}_{\text{smysl života}} \cdot \overbrace{d+e+f}^9 = 42$$


```

$$\underbrace{\overbrace{a+b+c}^6}_{\text{smysl života}} \cdot \overbrace{d+e+f}^9 = 42$$

Pro přidání matematických akcentů (např. **malých šipek** nebo **tildy**) ke jménům proměnných se hodí příkazy uvedené v tabulce 3.1 na straně 72. Svorky a tildy pokrývající několik znaků se vygenerují pomocí `\widetilde` a `\widehat`. Všimněte si rozdílu mezi `\hat` a `\widehat` a umístění `\bar` u proměnné s indexem. Znakem apostrof `'` získáme derivaci:

```


$$f(x) = x^2 \quad f'(x) = 2x$$


$$\hat{XY} \quad \widehat{XY}$$


$$\bar{x}_0 \quad \bar{x}$$


```

$$f(x) = x^2 \quad f'(x) = 2x \quad f''(x) = 2$$

$$\hat{XY} \quad \widehat{XY} \quad \bar{x}_0 \quad \bar{x}$$

Vektory často specifikujeme přidáním malých šipek nad jména proměnných pomocí příkazu `\vec`. Příkazy `\overrightarrow` a `\overleftarrow` jsou užitečné pro vyznačení vektoru z A do B :

```


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


```

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

Jména funkcí jako `\log` se často sází vzpřímeným písmem (tedy ne italkou, která je v matematickém módu používána implicitně), \LaTeX proto nabízí následující příkazy, kterými můžete vysázet názvy nejčastěji používaných funkcí:

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

```


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


```

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

Pro funkce, které v seznamu nejsou uvedeny, je potřeba použít příkaz `\DeclareMathOperator`. Tento příkaz má i hvězdičkovanou verzi pro funkce s limitami. `\DeclareMathOperator` funguje jen v preambuli dokumentu, zakomentované řádky v následujícím příkladu je tedy potřeba uvést tam.

```
%\DeclareMathOperator
% {\argh}{argh}
%\DeclareMathOperator
% *{\nut}{Nut}
\begin{equation*}
3\argh = 2\nut_{x=1}
\end{equation*}
```

$$3 \operatorname{argh} = 2 \operatorname{Nut}_{x=1}$$

Funkci modulo můžete vysázat buď příkazem `\bmod` (binární operátor „ $a \bmod b$ “) nebo příkazem `\pmod` (výrazy typu „ $x \equiv a \pmod{b}$ “):

```
$a\bmod b \ \
x\equiv a \pmod{b}$
```

$$a \bmod b$$

$$x \equiv a \pmod{b}$$

Zlomek se vysází pomocí příkazu `\frac{...}{...}`. V textovém stylu \LaTeX zlomky „zhuští“, aby se do řádků lépe vlezly. Zhuštění ve stylu `display` můžete explicitně zařídit pomocí příkazu `\tfrac`. Naopak, pomocí příkazu `\dfrac` můžete v textovém stylu zobrazit zlomek nezhuštěný. Pro krátké „zlomkové materiály“ často nejlépe vypadá, když zlomek jen naznačíme pomocí znaku lomítka ($1/2$):

```
Ve stylu display:
\begin{equation*}
3/8 \quad \frac{3}{8}
\quad \tfrac{3}{8}
\end{equation*}
```

Ve stylu display:

$$3/8 \quad \frac{3}{8} \quad \tfrac{3}{8}$$

```
V textovém stylu:
$1\frac{1}{2}$~hodiny \quad
$1\dfrac{1}{2}$~hodiny
```

V textovém stylu: $1\frac{1}{2}$ hodiny $1\frac{1}{2}$ hodiny

Zde použijeme příkaz `\partial` pro parciální derivace:

```
\begin{equation*}
\sqrt{\frac{x^2}{k+1}} \quad
x^{\frac{2}{k+1}} \quad
\frac{\partial^2 f}{\partial x^2}
\end{equation*}
```

$$\sqrt{\frac{x^2}{k+1}} \quad x^{\frac{2}{k+1}} \quad \frac{\partial^2 f}{\partial x^2}$$

Binomické koeficienty nebo podobné struktury můžete vysázat pomocí příkazu `\binom` definovaného v `amsmath`:

Pascalovo pravidlo říká, že

```
\begin{equation*}
\binom{n}{k} = \binom{n-1}{k}
+ \binom{n-1}{k-1}
\end{equation*}
```

Pascalovo pravidlo říká, že

$$\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}$$

U binárních relací může být výhodné umístit symboly nad sebe. Příkaz `\stackrel{#1}{#2}` umístí symbol uvedený jako #1 (velikostí písma stejnou jako pro horní indexy) nad symbol uvedený jako #2, jehož pozice zůstává „normální“.

```
\begin{equation*}
f_n(x) \stackrel{*}{\approx} 1
\end{equation*}
```

$$f_n(x) \overset{*}{\approx} 1$$

Symbol integrálu vygenerujeme pomocí `\int`, symbol sumy pomocí `\sum` a symbol součinu pomocí `\prod`. Horní a dolní limity se specifikují pomocí `^` a `_`, stejně jako u horních a dolních indexů:

```
\begin{equation*}
\sum_{i=1}^n \quad \iint \quad \prod_{\epsilon}
\end{equation*}
```

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

Příkaz `\substack` balíku `amsmath` umožňuje přesněji specifikovat umístění indexů u složitých výrazů:

```
\begin{equation*}
\sum^n_{\substack{0 < i < n \\ j \subseteq i}}
P(i, j) = Q(i, j)
\end{equation*}
```

$$\sum_{\substack{0 < i < n \\ j \subseteq i}}^n P(i, j) = Q(i, j)$$

L^AT_EX nabízí řadu symbolů pro **svorky** a další **oddělovače**, např. `[`, `<`, `||`, `↑`. Kulaté a hranaté závorky vysázíme uvedením příslušných znaků, složené závorky pak pomocí `\{`. Všechny ostatní oddělovače jsou generovány pomocí speciálních příkazů, např. `\updownarrow`.

```
\begin{equation*}
\{a, b, c\} \neq \{a, b, c\}
\end{equation*}
```

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


Pokud uvedete `\left` před otevíracím oddělovačem a `\right` před uzavíracím, L^AT_EX automaticky určí správnou velikost obou oddělovačů. Je ale potřeba každý `\left` uzavřít pomocí `\right`. Nechcete-li zobrazit žádný pravý oddělovač, použijte „neviditelný“ oddělovač `\left.`:

```
\begin{equation*}
1 + \left(\frac{1}{1-x^2}\right)^3 \quad
\left. \ddagger \right)
\end{equation*}
```

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

Někdy musíme určit výšku matematických oddělovačů explicitně, k čemuž slouží příkazy `\big`, `\Big`, `\bigg` a `\Bigg`, které lze uvést před většinou oddělovačů:

```
$$\Big((x+1)(x-1)\Big)^2$$
$\big( \Big( \bigg( \Bigg( \quad
\big\} \Big\} \bigg\} \Bigg\}
\quad
\big\! \Big\! \bigg\! \Bigg\!
\quad
\big\Downarrow \Big\Downarrow
\bigg\Downarrow \Bigg\Downarrow$
```

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


Všechny dostupné oddělovače jsou uvedeny v tabulce 3.8 na straně 75.

3.4. Příliš dlouhé rovnice: multiline

Je-li rovnice příliš dlouhá, musíme ji rozdělit do řádků, ale výsledek bude samozřejmě hůře čitelný, než kdyby se vešla na řádek jeden. Několik pravidel dodržенých při dělení rovnice do řádek může pomoci k lepší čitelnosti:

1. Obecně bychom měli rovnice dělit jen **před** rovnítkem nebo operátorem, přičemž první možnost je lepší.
2. Rozdělení rovnice před operátorem sčítání nebo odčítání je lepší než před operátorem násobení.
3. Je-li to možné, v jiných místech bychom rovnice dělit neměli.

Nejjednodušeji rovnici do řádek rozdělíme pomocí prostředí `multiline`⁶:

```
\begin{multiline}
a + b + c + d + e + f
+ g + h + i \\
= j + k + l + m + n
\end{multiline}
```

$$a + b + c + d + e + f + g + h + i \\ = j + k + l + m + n \quad (3.4)$$

Na rozdíl od prostředí `equation` můžeme pomocí `\` specifikovat řádkové zlomy (i vícenásobné). Hvězdičková verze `multiline*` potlačí vysázení čísla rovnice, podobně jako `equation*`.

⁶Definovaného v `amsmath`.

Prostředí `multline` se sice jednoduše používá, ale pomocí jiného prostředí `IEEEeqnarray`, viz sekce 3.5., často získáme lepší výsledky, např. v následující běžné situaci:

```
\begin{equation}
  a = b + c + d + e + f
    + g + h + i + j
    + k + l + m + n + o + p
  \label{eq:equation_too_long}
\end{equation}
```

$$a = b+c+d+e+f+g+h+i+j+k+l+m+n+o+p \quad (3.5)$$

Zde nám dělá problémy šířka pravé strany rovnice. Použijeme-li prostředí `multline`, vysázíme toto:

```
\begin{multline}
  a = b + c + d + e + f
    + g + h + i + j \\
    + k + l + m + n + o + p
\end{multline}
```

$$a = b + c + d + e + f + g + h + i + j \\ + k + l + m + n + o + p \quad (3.6)$$

To je samozřejmě lepší než (3.5), ale nevýhodou je, že rovnítko ztrácí přirozenou větší důležitost vzhledem k sčítacímu operátoru před k . Ještě lepší výsledek získáme pomocí prostředí `IEEEeqnarray`, které podrobně popíšeme v sekci 3.5.:

```
\begin{IEEEeqnarray}{rCl}
  a & = & b + c + d + e + f \\
    & + & g + h + i + j \nonumber \\
    & & + k + l + m + n + o + p \\
  \label{eq:dont_use_multline}
\end{IEEEeqnarray}
```

$$a = b + c + d + e + f + g + h + i + j \\ + k + l + m + n + o + p \quad (3.7)$$

Zde je druhý řádek horizontálně zarovnaný s prvním řádkem: $+$ před k je přesně pod b , takže pravá strana rovnice je jasně oddělena od levé.

3.5. Více rovnic

Obecně můžeme mít několik rovnic, z nichž některé se nevejdou na jednu řádku. Rovnice potřebujeme horizontálně zarovnat, aby jejich struktura byla dobře čitelná.

Před tím, než ukážeme naše návrhy na řešení, ukažme si, jak se věci dají udělat špatně. Tyto příklady demonstrují největší nevýhody některých běžných řešení.

3.5.1. Problémy s tradičními příkazy

K seskupení několika rovnic je možno použít prostředí `align`.⁷

```
\begin{align}
a &= b + c \\
&= d + e
\end{align}
```

$$a = b + c \quad (3.8)$$

$$= d + e \quad (3.9)$$

Pokud je ale některá rovnice příliš dlouhá, tento postup není vhodný:

```
\begin{align}
a &= b + c \\
&= d + e + f + g + h + \\
&\quad i + j + k + l \nonumber \\
&= m + n + o \\
&= p + q + r + s
\end{align}
```

$$a = b + c \quad (3.10)$$

$$= d + e + f + g + h + i + j + k + l$$

$$+ m + n + o \quad (3.11)$$

$$= p + q + r + s \quad (3.12)$$

Lépe by vypadalo, kdyby $+ m$ bylo umístěno pod d a ne pod rovnítkem. Samozřejmě bychom mohli $+ m$ posunout „ručním“ vložením potřebného množství mezer (např. pomocí `\hspace{...}`), ale přesné zarovnání tím nezískáme. Pozn. Takovýmto „ručním“ zásahům bychom se v každém případě měli vyhýbat.

Lepší je použít prostředí `eqnarray`:

```
\begin{eqnarray}
a &= & b + c \\
&= & d + e + f + g + h + \\
&\quad i + j + k + l \nonumber \\
& & + \quad m + n + o \\
&= & p + q + r + s
\end{eqnarray}
```

$$a = b + c \quad (3.13)$$

$$= d + e + f + g + h + i + j + k + l$$

$$+ m + n + o \quad (3.14)$$

$$= p + q + r + s \quad (3.15)$$

I toto prostředí má ale výrazné nevýhody:

- Mezery na obou stranách rovnítko jsou příliš velké. Zejména *nejdou* stejně velké jako mezery v prostředích `multline` a `equality`:

```
\begin{eqnarray}
a &= & a = a
\end{eqnarray}
```

$$a = a = a \quad (3.16)$$

⁷Prostředí `align` můžeme použít i na seskupení několika bloků funkcí vedle sebe. Pro toto zřídka používané řešení ale doporučujeme použít prostředí `IEEEeqnarray` s argumentem typu `{rCl+rCl}`.

- Rovnice se někdy překrývá se svým číslem, i když na levé straně je místa dost:

```
\begin{eqnarray}
a & = & b + c \\
& = & d + e + f + g + h^2 + i^2 + j
\label{eq:faultyeqnarray}
\end{eqnarray}
```

$$\begin{aligned} a &= b + c & (3.17) \\ &= d + e + f + g + h^2 + i^2 + j & (3.18) \end{aligned}$$

- Je zde sice k dispozici příkaz `\lefteqn`, který se dá použít v případě, že levá strana rovnice je příliš dlouhá:

```
\begin{eqnarray}
\lefteqn{a + b + c + d} \\
+ e + f + g + h \nonumber \\
& = & i + j + k + l + m \\
& = & n + o + p + q + r + s
\end{eqnarray}
```

$$\begin{aligned} a + b + c + d + e + f + g + h \\ &= i + j + k + l + m & (3.19) \\ &= n + o + p + q + r + s & (3.20) \end{aligned}$$

..., ale tento příkaz nemusí fungovat správně: pokud je pravá strana rovnice příliš krátká, rovnice nejsou správně vycentrovány:

```
\begin{eqnarray}
\lefteqn{a + b + c + d} \\
+ e + f + g \\
\nonumber \\
& = & i + j
\end{eqnarray}
```

$$\begin{aligned} a + b + c + d + e + f + g \\ &= i + j & (3.21) \end{aligned}$$

Navíc je obtížné změnit horizontální zarovnání rovnítka na druhém řádku.

Máme ale naštěstí lepší možnost...

3.5.2. Prostředí `IEEEeqnarray`

Prostředí `IEEEeqnarray` je mocné a bohatě konfigurovatelné. Představíme jen jeho základní použití, více informací lze nalézt v oficiálním manuálu,⁸ `IEEEeqnarray` je věnována příloha F.

Abychom mohli prostředí `IEEEeqnarray` používat, musíme nejdřív v našem dokumentu nahrát balík `IEEEtrantools`.⁹ Do preambule vašeho dokumentu přidejte řádku:

```
\usepackage[retainorgcmds]{IEEEtrantools}
```

⁸Soubor `IEEEtran_HOWTO.pdf`.

⁹Tento balík je dostupný na CTAN.

Silnou stránkou `IEEEeqnarray` je možnost zadat počet *sloupců* v poli rovnic. Obyčejně takto zadáme tři sloupce: `{rCl}`. První sloupec zarovnaný doprava, druhý vycentrovaný a s trochou mezer na obou stranách – proto bylo použito `C` místo `c` – a třetí sloupec zarovnaný doleva.

```
\begin{IEEEeqnarray}{rCl}
```

```
  a & = & b + c \\
```

```
  & = & d + e + f + g + h
```

```
  + i + j \nonumber\\
```

```
  && +\: k + l + m + n + o \\
```

```
  & = & p + q + r + s
```

```
\end{IEEEeqnarray}
```

$$a = b + c \quad (3.22)$$

$$= d + e + f + g + h + i + j + k + l + m + n + o \quad (3.23)$$

$$= p + q + r + s \quad (3.24)$$

Počet sloupců ale můžeme zadat jakýkoliv. Např. `c` znamená, že se použije jen jeden sloupec a položky budou vycentrované, nebo `{rCl1}` použije oproti předchozímu příkladu navíc jeden sloupec zarovnaný doleva. Kromě `l`, `c`, `r`, `L`, `C` a `R` pro položky v matematickém módu jsou k dispozici `s`, `t` a `u` pro položky v textovém módu zarovnané doleva, resp. vycentrované, resp. zarovnané doprava. Můžeme také přidat více mezer pomocí `.`, `/` a `?` (`?` přidá nejvíce).¹⁰

Všimněte si, že mezery okolo rovnítek jsou vysázeny správně (na rozdíl od `eqnarray`)!

3.5.3. Běžné použití

Nyní popíšeme, jak použít `IEEEeqnarray` v nejběžnějších situacích.

- Překrývá-li se rovnice se svým číslem (jako v (3.18)), můžeme použít příkaz

```
\IEEEeqnarraynumspace
```

Umístěním příkazu do příslušného řádku se zajistí, že celá soustava rovnic je posunuta o velikost čísla rovnice (při posunutí se vezme v úvahu velikost čísla!). Místo

```
\begin{IEEEeqnarray}{rCl}
```

```
  a & = & b + c \\
```

```
  & = & d + e + f + g + h
```

```
  + i + j + k \\
```

```
  & = & l + m + n
```

```
\end{IEEEeqnarray}
```

$$a = b + c \quad (3.25)$$

$$= d + e + f + g + h + i + j \quad (3.26)$$

$$= l + m + n \quad (3.27)$$

tak získáme

¹⁰Více typů mezer je uvedeno v sekci 3.9.1. a v oficiálním manuálu.

```
\begin{IEEEeqnarray}{rCl}
a & = & b + c \\
& & \\
& = & d + e + f + g + h \\
& + & i + j + k \\
\IEEEeqnarraynumspace \\
& = & l + m + n.
\end{IEEEeqnarray}
```

$$a = b + c \quad (3.28)$$

$$= d + e + f + g + h + i + j + k \quad (3.29)$$

$$= l + m + n. \quad (3.30)$$

- Je-li levá strana rovnice příliš dlouhá, prostředí `IEEEeqnarray` nabízí příkaz `\IEEEeqnarraymulticol`¹¹, který funguje ve všech situacích:

```
\begin{IEEEeqnarray}{rCl}
\IEEEeqnarraymulticol{3}{1}{
a + b + c + d + e + f
+ g + h
}\nonumber \\ \quad
& = & i + j \\
& = & k + l + m
\end{IEEEeqnarray}
```

$$a + b + c + d + e + f + g + h$$

$$= i + j \quad (3.31)$$

$$= k + l + m \quad (3.32)$$

Použití je stejné jako u příkazu `\multicolumns` v prostředí `tabular`. První argument, `{3}`, specifikuje, že se mají zkombinovat tři sloupce do jednoho, který bude zarovnaný vlevo, `{1}`.

Všimněte si, že úpravou příkazu `\quad` jednoduše upravíme hloubku rovnítek,¹² *např.*

```
\begin{IEEEeqnarray}{rCl}
\IEEEeqnarraymulticol{3}{1}{
a + b + c + d + e + f
+ g + h
}\nonumber \\ \quad \quad \quad
& = & i + j \\
& = & k + l + m
\end{IEEEeqnarray}
```

$$a + b + c + d + e + f + g + h$$

$$= i + j \quad (3.33)$$

$$= k + l + m \quad (3.34)$$

- Je-li rovnice rozdělena do více řádek, `LATEX` interpretuje první `+` nebo `-` na dalším řádku jako znaménko (a ne jako operátor). Je proto nutné přidat za operátor mezeru. Místo...

¹¹Snažíme se vyhnout špatně fungujícímu příkazu `\lefteqn`.

¹²Myslím, že vzdálenost jeden quad vypadá dobře ve většině případech.

```
\begin{IEEEeqnarray}{rCl}
a & = & b + c \\
& = & d + e + f + g + h
+ i + j + k \nonumber \\
& = & l + m + n + o \\
& = & p + q + r + s
\end{IEEEeqnarray}
```

$$a = b + c \quad (3.35)$$

$$= d + e + f + g + h + i + j + k \\ + l + m + n + o \quad (3.36)$$

$$= p + q + r + s \quad (3.37)$$

... bychom tedy měli psát

```
\begin{IEEEeqnarray}{rCl}
a & = & b + c \\
& = & d + e + f + g + h
+ i + j + k \nonumber \\
& +\!:\! & l + m + n + o \\
& = & p + q + r + s
\end{IEEEeqnarray}
```

$$a = b + c \quad (3.38)$$

$$= d + e + f + g + h + i + j + k \\ + l + m + n + o \quad (3.39)$$

$$= p + q + r + s \quad (3.40)$$

(Porovnejte mezeru mezi + a l.)

Upozornění: v některých situacích \LaTeX pozná, že + nebo – musí být operátor (a ne znaménko), vloží správnou mezeru a my nemusíme ručně nic měnit. \LaTeX (správně) odvodí, že dané + nebo – je operátor (a ne znaménko) pokud je uvedeno před:

- jménem operátoru, např. \log , \sin , \det , \max , *atd.*,
- integrálem \int nebo sumou \sum ,
- závorkou s proměnlivou velikostí, zapsanou pomocí \left nebo \right (ne ale normální závorkou nebo závorkou s pevnou velikostí, např. $\big()$).

Proto:

▷ *Kdykoliv zalomíte řádku, rychle zkontrolujte, jestli jsou mezery v pořádku!*

- Nemá-li konkrétní řádek obsahovat číslo rovnice, můžeme vysázení tohoto čísla potlačit pomocí \nonumber (nebo pomocí \IEEEnonumber . Pokud navíc na takovémto řádku uvedeme $\label{eq:...}$, pak je tento label předán první z následujících rovnic, jejíž číslo není potlačeno. Label se ale doporučuje uvádět raději těsně před zlomem řádku, \backslash , nebo koncem rovnice, do které řádek patří. Kromě lepší čitelnosti \LaTeX ového zdrojového kódu tak zabráníme chybě při překladu v situaci, kdy za labelem je uvedeno \IEEEmulticol .
- Máme k dispozici i hvězdičkovanou verzi, kde jsou všechna čísla rovnic potlačena. Číslo řádku zde můžeme explicitně zobrazit pomocí příkazu \IEEEyesnumber :

```
\begin{IEEEeqnarray*}{rCl}
  a & = & b + c \\
  & = & d + e \IEEEyesnumber\
  & = & f + g
\end{IEEEeqnarray*}
```

$$\begin{aligned} a &= b + c \\ &= d + e \\ &= f + g \end{aligned} \quad (3.41)$$

- Pomocí `\IEEEyessubnumber` můžeme také použít „podčísla“:

```
\begin{IEEEeqnarray}{rCl}
  a & = & b + c \\
  \IEEEyessubnumber\
  & = & d + e \\
  \nonumber\
  & = & f + g \\
  \IEEEyessubnumber
\end{IEEEeqnarray}
```

$$\begin{aligned} a &= b + c & (3.42a) \\ &= d + e \\ &= f + g & (3.42b) \end{aligned}$$

3.6. Pole a matice

K vysázení **polí** používáme prostředí `array`. Trochu se podobá prostředí `tabular`. Příkazem `\\` rozdělíme řádky:

```
\begin{equation*}
  \mathbf{X} = \left(
    \begin{array}{ccc}
      x_1 & x_2 & \dots \\
      x_3 & x_4 & \dots \\
      \vdots & \vdots & \ddots
    \end{array}
  \right)
\end{equation*}
```

$$\mathbf{X} = \begin{pmatrix} x_1 & x_2 & \dots \\ x_3 & x_4 & \dots \\ \vdots & \vdots & \ddots \end{pmatrix}$$

Prostředí `array` můžeme použít k vysázení definic funkcí, které mají několik variant. Je pak potřeba použít `.` jakožto skrytý pravý oddělovač:

```
\begin{equation*}
  |x| = \left\{
    \begin{array}{rl}
      -x & \text{if } x < 0, \\
      0 & \text{if } x = 0, \\
      x & \text{if } x > 0.
    \end{array}
  \right.
\end{equation*}
```

$$|x| = \begin{cases} -x & \text{if } x < 0, \\ 0 & \text{if } x = 0, \\ x & \text{if } x > 0. \end{cases}$$

Stejnou věc můžeme jednodušeji vysázet pomocí prostředí `cases` též z balíku `amsmath`:

```

\begin{equation*}
|x| =
\begin{cases}
-x & \text{if } x < 0, \\
0 & \text{if } x = 0, \\
x & \text{if } x > 0.
\end{cases}
\end{equation*}

```

$$|x| = \begin{cases} -x & \text{if } x < 0, \\ 0 & \text{if } x = 0, \\ x & \text{if } x > 0. \end{cases}$$

Pomocí `array` můžeme vysázet i matice, ale `amsmath` poskytuje lepší možnost – několik verzí prostředí `matrix`. K dispozici je šest verzí tohoto prostředí, které se liší v použitých oddělovačích: `matrix` (žádný oddělovač), `pmatrix` (`()`), `bmatrix` (`[]`), `Bmatrix` (`{}`), `vmatrix` (`|`) a `Vmatrix` (`||`).

Na rozdíl od `array` není třeba zadávat počet sloupců, jejich maximální počet je konfigurovatelný, implicitně deset.

```

\begin{equation*}
\begin{matrix}
1 & 2 \\
3 & 4
\end{matrix} \quad \text{\quad}
\begin{bmatrix}
p_{11} & p_{12} & \dots & p_{1n} \\
p_{21} & p_{22} & \dots & p_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
p_{m1} & p_{m2} & \dots & p_{mn}
\end{bmatrix}
\end{equation*}

```

$$\begin{matrix} 1 & 2 \\ 3 & 4 \end{matrix} \quad \begin{bmatrix} p_{11} & p_{12} & \dots & p_{1n} \\ p_{21} & p_{22} & \dots & p_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{m1} & p_{m2} & \dots & p_{mn} \end{bmatrix}$$

3.7. Mezery v matematickém módu

Jestliže nejsme spokojeni s mezerami, jaké `LATEX` použil při vysázení matematického vzorce, můžeme je upravit vložení speciálních „mezerových“ příkazů: `\`, pro $\frac{3}{18}$ quad (`\`), `\:` pro $\frac{4}{18}$ quad (`\:`) a `\;` pro $\frac{5}{18}$ quad (`\;`). Zpětné lomítko následované mezerou, `_`, vygeneruje mezeru střední velikosti zhruba velikosti mezislovní mezery. Příkazy `\quad` (`\quad`) a `\qquad` (`\qquad`) vygenerují mezery větší. Velikost `\quad` je rovna šířce písmene ‘M’ aktivního fontu. `\!` vygeneruje zápornou mezeru $-\frac{3}{18}$ quad (`\!`).

Všimněte si, že ‘d’ označující diferenciál se tradičně sází vzpřímeně:

```
\begin{equation*}
\int_1^2 \ln x \mathrm{d}x
\quad
\int_1^2 \ln x \, \mathrm{d}x
\end{equation*}
```

$$\int_1^2 \ln x dx \quad \int_1^2 \ln x dx$$

V následujícím příkladu definujeme nový příkaz `\ud`, který vygeneruje “d” (všimněte si mezery `\` před `d`), abychom si ušetřili psaní.

Příkaz `\newcommand` se umísťuje obvykle do preambule dokumentu.

```
\newcommand{\ud}{\, \mathrm{d}}
```

```
\begin{equation*}
\int_a^b f(x)\ud x
\end{equation*}
```

$$\int_a^b f(x) dx$$

Chcete-li vysázet vícenásobné integrály, zjistíte, že mezery mezi symboly jednotlivých integrálů jsou příliš široké. Můžete je buď upravit pomocí `\!`, nebo použít příkazy `\iint`, `\iiint`, `\iiiiint` nebo `\idotsint` definované v balíku `amsmath`.

```
\newcommand{\ud}{\, \mathrm{d}}
```

```
\begin{IEEEeqnarray*}{c}
\int\int f(x)g(y)
\ud x \ud y \ \ \
\int\!\!\!\!\int
f(x)g(y) \ud x \ud y \ \ \
\iint f(x)g(y) \ud x \ud y
\end{IEEEeqnarray*}
```

$$\int \int f(x)g(y) dx dy$$

$$\iint f(x)g(y) dx dy$$

$$\iint f(x)g(y) dx dy$$

Další informace najdete v dokumentu `testmath.tex` (distribuovaném spolu s `AMS-LATEX`) nebo kapitole osm knihy *The L^AT_EX Companion* [3].

3.7.1. Neviditelné výrazy

Při vertikálním zarovnávání textu pomocí `^` a `_` občas potřebujeme upravit horizontální mezery implicitně `LATEX`em použité. K tomu se hodí příkaz `\phantom`, pomocí kterého vložíme mezeru, která má stejné rozměry jako by byly rozměry příslušného textu uvedeného jako parametr příkazu. Typické použití je ukázáno v následujícím příkladu:

```
\begin{equation*}
{}^{14}_6\text{C}
\quad
\text{versus}
\quad
{}^{14}_6\phantom{C}
\end{equation*}
```

$${}^{14}_6\text{C} \quad \text{versus} \quad {}^{14}_6\text{C}$$

Poznámka: Pokud sázíte větší množství chemických výrazů (např. izotopů z předchozího příkladu), můžete použít balík mhchem.

3.8. Hrátky s matematickými fonty

Různé matematické fonty jsou uvedeny v tabulce 3.14 na straně 76.

```

 $\Re$   $\mathcal{R}$   $\mathfrak{R}$   $\mathbb{R}$ 

```

\Re	\mathcal{R}	\mathfrak{R}	\mathbb{R}
-------	---------------	----------------	--------------

Poslední dva fonty vyžadují `amssymb` nebo `amsfonts`.

Občas potřebujete \LaTeX u zadat správnou velikost fontu. V matematickém módu se to dělá pomocí následujících čtyř příkazů:

```

 $\displaystyle (123)$ ,  $\textstyle (123)$ ,  $\scriptstyle (123)$ 
a  $\scriptscriptstyle (123)$ .

```

Jestliže je \sum uvedeno jako součást zlomku, bude příslušný symbol vysázen v textovém stylu, což můžete změnit, např. pomocí `\displaystyle`.

```


$$P = \frac{\sum_{i=1}^n (x_i - x)(y_i - y)}{\left[ \sum_{i=1}^n (x_i - x)^2 \sum_{i=1}^n (y_i - y)^2 \right]^{1/2}}$$


```

$P = \frac{\sum_{i=1}^n (x_i - x)(y_i - y)}{\left[\sum_{i=1}^n (x_i - x)^2 \sum_{i=1}^n (y_i - y)^2 \right]^{1/2}}$
--

Poznámka: Explicitní měnění stylů může ovlivnit zobrazení velkých operátorů a limit.

3.8.1. Tučné symboly

Vysázení tučných symbolů v \LaTeX u je docela obtížné. Pravděpodobně to byl záměr, protože amatérští sazeči je používají zbytečně často. Příkazem `\mathbf` sice můžeme nastavit tučné písmo, ale jen pro vzpřímené písmo a ne pro italiky, se kterou při sazbě matematiky pracujeme především. Navíc se nezmění tučnost malých řeckých písmen. Jinou možností je použít příkaz `\boldmath`. Je nutné ho uvést *mimo matematický mód*. Příkaz `\boldmath` funguje i pro symboly:

```

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

```

μ , M	$\mathbf{\mu}$, \mathbf{M}	$\boldmath{\mu}$, \boldmath{M}
-------------	-------------------------------	-----------------------------------

Balíky `amsbsy` (součást `amsmath`) a `bm` (kolekce balíků `tools`) obsahují příkaz `\boldsymbol`, zde jedna ukázka:

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

μ, M	$\boldsymbol{\mu}, \boldsymbol{M}$
----------	------------------------------------

3.9. Věty, lemmata, ...

Při tvorbě matematických dokumentů pravděpodobně budete potřebovat způsob, jak vysázet lemmata, definice, axiomy a podobné struktury.

<code>\newtheorem{name}[counter]{text}[section]</code>
--

Argument *name* je krátké jméno, které identifikuje daný *teorém*. Argument *text* je skutečné jméno teorému, které bude vysázeno.

Argumenty v hranatých závorkách jsou nepovinné. Oba specifikují číslování, které se pro teorém použije. Argumentem *counter* můžete specifikovat jméno (*name*) dříve deklarovaného teorému. Nový teorém pak bude číslován ve stejné posloupnosti. Argument *section* umožňuje specifikovat jednotku, o kterou budou čísla teorémů zvětšována.

Příkaz `\newtheorem` uvedený v preambuli vašeho dokumentu je \LaTeX em proveden a vy potom v dokumentu můžete tento příkaz používat.

```
\begin{name}[text]
Toto je text mé zajímavé věty.
\end{name}
```

Balík `amsthm` poskytuje příkaz `\theoremstyle{style}`, pomocí kterého můžete vybrat jeden ze tří předdefinovaných stylů: `definition` (titulek teorému tučným písmem, tělo teorému normálním – vzpřímeným – písmem), `plain` (titulek tučným, tělo italikou) nebo `remark` (titulek italikou, tělo vzpřímeným písmem). Balík `amsthm` je součástí $\mathcal{A}\mathcal{M}\mathcal{S}$ - \LaTeX u.

Dost už teorie. Následující příklady by měly vyjasnit všechny pochybnosti ohledně použití prostředí `\newtheorem`.

Nejdříve nadefinujeme teorémy:

```
\theoremstyle{definition} \newtheorem{law}{Právo}
\theoremstyle{plain}      \newtheorem{jury}[law]{Porota}
\theoremstyle{remark}     \newtheorem*{marg}{Margaret}
```

```

\begin{law} \label{law:box}
Neskrývejte se v lavici svědků!
\end{law}
\begin{jury}[12 soudců]
Mohl jste to být vy! Dejte
si pozor a dodržujte
zákon~\ref{law:box}.\end{jury}
\begin{marg}Ne, ne, ne!\end{marg}

```

Právo 1. Neskrývejte se v lavici svědků!

Porota 2 (12 soudců). *Mohl jste to být vy! Dejte si pozor a dodržujte zákon 1.*

Margaret. Ne, ne, ne!

Teorém „Porota“ používá stejný čítač jako teorém „Právo“, takže číslo tohoto čítače budou zvyšovat všechna použití jak věty „Porota“, tak věty „Právo“. Argument v hranatých závorkách specifikuje titulek teorému.

```

\newtheorem
{mur}{Murphy}[section]

\begin{mur} Jestliže se něco
dá udělat dvěma nebo více
způsoby a jeden z nich vede
ke katastrofě, někdo si tento
způsob zvolí.\end{mur}

```

Murphy 3.9.1. Jestliže se něco dá udělat dvěma nebo více způsoby a jeden z nich vede ke katastrofě, někdo si tento způsob zvolí.

„Murphyho“ teorému je přiděleno číslo, které se vztahuje k číslu aktuální sekce. Je možno použít i jinou „jednotku“, tedy místo sekce použít např. kapitolu nebo podsekcí.

Jestliže si chcete teorémy přizpůsobit do posledního detailu, můžete použít balík `ntheorem`.

3.9.1. Důkazy a symbol „konec důkazu“

Balík `amsthm` obsahuje i definici prostředí `proof`.

```

\begin{proof}
Triviální, použijte
\begin{equation*}
E=mc^2.
\end{equation*}
\end{proof}

```

Důkaz. Triviální, použijte

$$E = mc^2.$$

□

Příkazem `\qedhere` můžete symbol „konec důkazu“ posunout – užitečné v případě, kdy by jinak tento symbol byl sám vysázen na novém řádku.

```

\begin{proof}
Triviální, použijte
\begin{equation*}
E=mc^2. \qedhere
\end{equation*}
\end{proof}

```

Důkaz. Triviální, použijte

$$E = mc^2.$$

□

Bohužel ale tato úprava nebude fungovat pro `IEEEeqnarray`:

```
\begin{proof}
  Toto je důkaz, který končí
  soustavou rovnic:
  \begin{IEEEeqnarray*}{rCl}
    a & = & b + c & \\
    & = & d + e. & \qedhere
  \end{IEEEeqnarray*}
\end{proof}
```

Důkaz. Toto je důkaz, který končí soustavou rovnic:

$$\begin{aligned} a &= b + c \\ &= d + e. \quad \square \end{aligned}$$

Důvodem je vnitřní struktura `IEEEeqnarray`: k oběma okrajům soustavy rovnic je připojen neviditelný sloupec obsahující natahovací mezery. Tím `IEEEeqnarray` zajistí, že soustava bude horizontálně vycentrována. Příkaz `\qedhere` by bylo potřeba umístit *vně* této natahovací mezery, ale to nejde, protože oba „neviditelné sloupce“ jsou uživateli nepřístupné.

Máme ale jednoduchou možnost nápravy: natahovací sloupce nadefinujeme sami!

```
\begin{proof}
  Toto je důkaz, který končí
  soustavou rovnic:
  \begin{IEEEeqnarray*}{+rCl+x*}
    a & = & b + c & \\
    & = & d + e. & \qedhere
  \end{IEEEeqnarray*}
\end{proof}
```

Důkaz. Toto je důkaz, který končí soustavou rovnic:

$$\begin{aligned} a &= b + c \\ &= d + e. \quad \square \end{aligned}$$

Všimněte si, že `+v {+rCl+x*}` značí natahovací mezery, jednu nalevo od rovnic, kterou, pokud není specifikována, určí `IEEEeqnarray` automaticky!, a druhou napravo. Napravo, za natahovací sloupec, umístíme prázdný sloupec, `x`. Ten budeme potřebovat jen na posledním řádku, na kterém do tohoto sloupce umístíme příkaz `\qedhere`. Konečně, specifikujeme `*`, což značí nulovou mezeru, která zabrání, aby `IEEEeqnarray` přidalo další natahovací mezeru `+`!

S číslováním rovnic máme podobný problém. Když porovnáte

```
\begin{proof}
  Toto je důkaz, který končí
  číslovanou rovnicí:
  \begin{equation}
    a = b + c.
  \end{equation}
\end{proof}
```

Důkaz. Toto je důkaz, který končí číslovanou rovnicí:

$$a = b + c. \quad (3.43)$$

□

s následujícím příkladem

```

\begin{proof}
  Toto je důkaz, který končí
  číslovanou rovnicí:
  \begin{equation}
    a = b + c. \quad \text{\qedhere}
  \end{equation}
\end{proof}

```

Důkaz. Toto je důkaz, který končí číslovanou rovnicí:

$$a = b + c. \quad (3.44) \quad \square$$

všimnete si, že v druhé verzi je (korektně) \square mnohem blíže k rovnici než v první verzi.

Podobně, správný způsob vložení symbolu „konec důkazu“ na konec soustavy rovnic je tento:

```

\begin{proof}
  Toto je důkaz, který končí
  soustavou rovnic:
  \begin{IEEEeqnarray}{+rCl+x*}
    a & = & b + c \\
    & & d + e. \\
    & & \text{\qedhere\nonumber}
  \end{IEEEeqnarray}
\end{proof}

```

Důkaz. Toto je důkaz, který končí soustavou rovnic:

$$a = b + c \quad (3.45)$$

$$= d + e. \quad (3.46) \quad \square$$

Na rozdíl od (nesprávného způsobu):

```

\begin{proof}
  Toto je důkaz, který končí
  soustavou rovnic:
  \begin{IEEEeqnarray}{rCl}
    a & = & b + c \\
    & & d + e.
  \end{IEEEeqnarray}
\end{proof}

```

Důkaz. Toto je důkaz, který končí soustavou rovnic:

$$a = b + c \quad (3.47)$$

$$= d + e. \quad (3.48) \quad \square$$

3.10. Seznam matematických symbolů

Následující tabulky ukazují všechny symboly obvykle přístupné v *matematickém módu*.

Abyste mohli používat symboly uvedené v tabulkách 3.12 až 3.19,¹³ musíte v preambuli vašeho dokumentu nahrát balík `amssymb` a na vašem systému musí být nainstalovány matematické fonty \mathcal{AMS} . Pokud balík nebo fonty \mathcal{AMS} nainstalovány nemáte, podívejte se do `CTAN:macros/latex/required/amslatex`. Ještě delší seznam matematických symbolů lze získat na `CTAN:info/symbols/comprehensive`.

Tabulka 3.1: Akcenty matematického módu.

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

Tabulka 3.2: Řecká písmena.

„Velká písmena“ pro část řecké abecedy jsou stejná jako pro latinkovou abecedu, např. A jako „velké“ α nebo B jako „velké“ β , a v tabulce proto nejsou uvedena.

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

¹³Tyto tabulky byly odvozeny ze souboru `symbols.tex` autora Davida Carlisleho a následně výrazně změněny podle návrhů Josefa Tkadlece.

Tabulka 3.3: Binární relace.

Následující symboly lze „znegovat“, uvedete-li před ně příkaz `\not`.

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

^aPro použití tohoto symbolu je nutno nahrát balík `latexsym`.

Tabulka 3.4: Binární operátory.

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

Tabulka 3.5: VELKÉ operátory.

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

Tabulka 3.6: Šipky.

\leftarrow	<code>\leftarrow</code> nebo <code>\gets</code>	\longleftarrow	<code>\longleftarrow</code>
\rightarrow	<code>\rightarrow</code> nebo <code>\to</code>	\longrightarrow	<code>\longrightarrow</code>
\leftrightarrow	<code>\leftrightarrow</code>	\longleftrightarrow	<code>\longleftrightarrow</code>
\Leftarrow	<code>\Leftarrow</code>	\Lleftarrow	<code>\Lleftarrow</code>
\Rightarrow	<code>\Rightarrow</code>	\Rrightarrow	<code>\Rrightarrow</code>
\Leftrightarrow	<code>\Leftrightarrow</code>	\Llongleftrightarrow	<code>\Llongleftrightarrow</code>
\mapsto	<code>\mapsto</code>	\longmapsto	<code>\longmapsto</code>
\hookrightarrow	<code>\hookrightarrow</code>	\hookleftarrow	<code>\hookleftarrow</code>
\lleftarrow	<code>\lleftarrow</code>	\rightharpoonup	<code>\rightharpoonup</code>
\lrightarrow	<code>\lrightarrow</code>	\rightharpoonleft	<code>\rightharpoonleft</code>
\rightleftharpoons	<code>\rightleftharpoons</code>	\iff	<code>\iff</code> (větší mezery)
\uparrow	<code>\uparrow</code>	\downarrow	<code>\downarrow</code>
\Uparrow	<code>\Uparrow</code>	\Downarrow	<code>\Downarrow</code>
\nearrow	<code>\nearrow</code>	\searrow	<code>\searrow</code>
\swarrow	<code>\swarrow</code>	\nwarrow	<code>\nwarrow</code>
\leadsto	<code>\leadsto</code> ^a		

^aPro použití tohoto symbolu je nutno nahrát balík `latexsym`.

Tabulka 3.7: Šipky a akcenty.

\overrightarrow{AB}	<code>\overrightarrow{AB}</code>	$\underline{\overrightarrow{AB}}$	<code>\underrightarrow{AB}</code>
\overleftarrow{AB}	<code>\overleftarrow{AB}</code>	$\underline{\overleftarrow{AB}}$	<code>\underleftarrow{AB}</code>
\overleftrightarrow{AB}	<code>\overleftrightarrow{AB}</code>	$\underline{\overleftrightarrow{AB}}$	<code>\underleftrightarrow{AB}</code>

Tabulka 3.8: Oddělovače.

(())	↑	\uparrow
[[or \lbrack]] or \rbrack	↓	\downarrow
{	\{ or \lbrace	}	\} or \rbrace	↕	\updownarrow
<	\langle	>	\rangle	⇑	\Uparrow
	or \vert		\ or \Vert	⇓	\Downarrow
/	/	\	\backslash	↕	\Updownarrow
⌊	\lfloor	⌋	\rfloor		
⌈	\lceil	⌉	\rceil		

Tabulka 3.9: Velké oddělovače.

(\lgroup)	\rgroup	⎵	\lmoustache
	\arrowvert		\Arrowvert		\bracevert
)	\rmoustache				

Tabulka 3.10: Různé symboly.

...	\dots	...	\cdots	:	\vdots	⋯	\ddots
\hbar	\hbar	\imath	\imath	\jmath	\jmath	ℓ	\ell
\Re	\Re	\Im	\Im	\aleph	\aleph	\wp	\wp
\forall	\forall	\exists	\exists	\mho	\mho ^a	∂	\partial
'	'	'	\prime	\emptyset	\emptyset	∞	\infty
∇	\nabla	\triangle	\triangle	\square	\Box ^a	\diamond	\Diamond ^a
\perp	\bot	\top	\top	\angle	\angle	\surd	\surd
\diamond	\diamondsuit	\heartsuit	\heartsuit	\clubsuit	\clubsuit	\spadesuit	\spadesuit
\neg	\neg or \lnot	\flat	\flat	\natural	\natural	\sharp	\sharp

^aPro použití tohoto symbolu je nutno nahrát balík latexsym.

Tabulka 3.11: Nematematické symboly.

Tyto symboly lze použít i v textovém módu.

†	\dag	§	\S	©	\copyright	®	\textregistered
‡	\ddag	¶	\P	£	\pounds	%	\%

Tabulka 3.12: \mathcal{AMS} oddělovače.

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

Tabulka 3.13: \mathcal{AMS} Řečtina a Hebrejština.

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

Tabulka 3.14: Matematické abecedy.

Další matematické fonty jsou uvedeny v tabulce 6.4 na straně 121.

Příklad	Příkaz	Vyžadovaný balík
$ABCDEabcde1234$	<code>\mathrm{ABCDE abcde 1234}</code>	
$ABCDEabcde1234$	<code>\mathit{ABCDE abcde 1234}</code>	
$ABCDEabcde1234$	<code>\mathnormal{ABCDE abcde 1234}</code>	
$ABCDE$	<code>\mathcal{ABCDE abcde 1234}</code>	
$\mathcal{A}\mathcal{B}\mathcal{C}\mathcal{D}\mathcal{E}$	<code>\mathscr{ABCDE abcde 1234}</code>	mathrsfs
$\mathfrak{A}\mathfrak{B}\mathfrak{C}\mathfrak{D}\mathfrak{E}\mathfrak{a}\mathfrak{b}\mathfrak{c}\mathfrak{d}\mathfrak{e}\mathfrak{1}\mathfrak{2}\mathfrak{3}\mathfrak{4}$	<code>\mathfrak{ABCDE abcde 1234}</code>	amsfonts or amssymb
$\mathbb{A}\mathbb{B}\mathbb{C}\mathbb{D}\mathbb{E}\mathbb{a}\mathbb{b}\mathbb{c}\mathbb{d}\mathbb{e}\mathbb{1}\mathbb{2}\mathbb{3}\mathbb{4}$	<code>\mathbb{ABCDE abcde 1234}</code>	amsfonts or amssymb

Tabulka 3.15: \mathcal{AMS} Binární operátory.

$\dot{+}$	<code>\dotplus</code>	\cdot	<code>\centerdot</code>		
\ltimes	<code>\ltimes</code>	\rtimes	<code>\rtimes</code>	\div	<code>\divideontimes</code>
\cup	<code>\doublecup</code>	\cap	<code>\doublecap</code>	\smallsetminus	<code>\smallsetminus</code>
\veebar	<code>\veebar</code>	$\bar{\wedge}$	<code>\barwedge</code>	$\bar{\bar{\wedge}}$	<code>\doublebarwedge</code>
\boxplus	<code>\boxplus</code>	\boxminus	<code>\boxminus</code>	\ominus	<code>\circleddash</code>
\boxtimes	<code>\boxtimes</code>	\boxdot	<code>\boxdot</code>	\odot	<code>\circledcirc</code>
\intercal	<code>\intercal</code>	\circledast	<code>\circledast</code>	\times	<code>\rightthreetimes</code>
\curlyvee	<code>\curlyvee</code>	\curlywedge	<code>\curlywedge</code>	\times	<code>\leftthreetimes</code>

Tabulka 3.16: $\mathcal{A}\mathcal{M}\mathcal{S}$ Binární relace.

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

Tabulka 3.17: \mathcal{AMS} Šipky.

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

Tabulka 3.18: \mathcal{AMS} Znegované binární relace a šipky.

\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
\precneqq	\succneqq	\nvDash
\precnsim	\succnsim	\nVdash
\precnapprox	\succnapprox	\nVDash
\subsetneq	\supsetneq	\triangleleft
\varsubsetneq	\varsupsetneq	\triangleright
\subsetneqq	\supsetneqq	\trianglelefteq
\subsetneqq	\supsetneqq	\trianglerighteq
\nleftarrow	\nrightarrow	\leftrightharrow
\nLeftarrow	\nRightarrow	\Leftrightarrow

Tabulka 3.19: \mathcal{AMS} Různé.

\hbar	\hslash	\Bbbk
\square	\blacksquare	\textcircled{S}
\vartriangle	\blacktriangle	\complement
\triangledown	\blacktriangledown	\Game
\lozenge	\blacklozenge	\bigstar
\angle	\measuredangle	\backprime
\diagup	\diagdown	\varnothing
\nexists	\Finv	\mho
\eth	\sphericalangle	

Kapitola 4.

Speciality

Při tvorbě rozsáhlých dokumentů vám \LaTeX může pomoci řadou „specialit“, např. vygenerováním indexu (též rejstříku) nebo udržováním bibliografie (též seznam zdrojů). Výrazně úplnější popis specialit a rozšíření, která \LaTeX poskytuje je uveden v *\LaTeX Manual* [1] a *The \LaTeX Companion* [3].

4.1. Vkládání Encapsulated Postscript grafiky

\LaTeX umožňuje pracovat s „plovoucími objekty“, jako jsou obrázky nebo tabulky, pomocí prostředí `figure` a `table`.

Grafiku lze několika způsoby generovat přímo buď v základním \LaTeX u nebo v některém jeho rozšiřujícím balíku. Několik těchto balíků je popsáno v kapitole 5. Více informací je uvedeno v *The \LaTeX Companion* [3] a *\LaTeX Manual* [1].

Mnohem jednodušší ale je vygenerovat grafiku pomocí specializovaného programu¹ a potom ji už jen vložit do \LaTeX ového dokumentu. Vložení se dá udělat mnoha způsoby, tento úvod ale rozebere jen vkládání grafiky ve formátu Encapsulated POSTSCRIPT (EPS), protože je jednoduché a široce používané. Abyste si dokument s vloženou EPS grafikou mohli vytisknout, budete potřebovat postscriptovou tiskárnu nebo program GHOSTSCRIPT.²

Širokou množinu příkazů pro vkládání grafiky poskytuje balík `graphicx` od D. P. Carlisle. Je součástí celé rodiny balíků „graphic tools“.³

Pokud pracujete na systému s postscriptovou tiskárnou a s nainstalovaným balíkem `graphicx`, můžete obrázky do svého dokumentu vkládat v následujících krocích:

¹Např. XFig, Gnuplot, Gimp, Xara X, ...

²Dostupný v CTAN://tex-archive/support/ghostscript. Uživatelé Windows a OS/2 můžou použít i program GSVIEW.

³CTAN://tex-archive/macros/latex/required/graphics

1. Exportujte vámi vygenerovaný soubor s grafikou do formátu EPS.⁴
2. Nahrajte balík `graphicx` v preambuli vstupního souboru pomocí

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

kde *driver* je jméno programu konvertujícího z formátu dvi do formátu postscript. Nejpoužívanější z těchto programů je `dvips`. Jméno konverzního programu je potřeba zadat z toho důvodu, že vkládání grafiky do TeXu není standardizováno. Zná-li jméno driveru, může balík `graphicx` vložit do `.dvi` souboru informaci o grafice způsobem, kterému bude driver rozumět a tuto grafiku správně vloží.

3. Pomocí příkazu

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

vložte *file* (příslušný grafický soubor) do svého dokumentu. V nepovinném parametru můžete předat seznam položek *klíč-hodnota*, pomocí nichž lze změnit šířku, výšku nebo rotaci vkládané grafiky. Nejdůležitější klíče jsou uvedeny v tabulce 4.1.

Tabulka 4.1: Význam jmen klíčů balíku `graphicx`.

<code>width</code>	Upraví grafiku tak, aby měla uvedenou šířku.
<code>height</code>	Upraví grafiku tak, aby měla uvedenou výšku.
<code>angle</code>	Natočí grafiku proti směru hodinových ručiček.
<code>scale</code>	Upraví velikost grafiky.

Následující příklad by měl vyjasnit předchozí popis:

```
\begin{figure}
\centering
\includegraphics[angle=90,
width=0.5\textwidth]{test}
\caption{This is a test.}
\end{figure}
```

⁴Pokud nemáte nainstalován žádný program, který by uměl konvertovat grafiku do formátu EPS, můžete si zkusit nainstalovat ovladač postscriptové tiskárny (např. Apple LaserWriter) a vytisknout pomocí tohoto ovladače soubor s grafikou do souboru. Je pravděpodobné, že „vytištěný“ soubor bude ve formátu EPS. Pozor na to, že EPS nesmí obsahovat více než jednu stránku. Ovladače některých jiných tiskáren lze nakonfigurovat tak, aby EPS formát generovaly také.

Tento kód vloží grafiku uloženou v souboru `test.eps`. Grafika je *nejdřív* otočena o devadesát stupňů a *potom* je její velikost změněna tak, aby její šířka byla polovinou standardní šířky odstavcového řádku (výška je upravena ve stejném poměru jako šířka). Výšku a šířku je také možné uvést v absolutních jednotkách. Více informací je uvedeno v tabulce 6.5 na straně 125. Další informace naleznete v [9] a [13].

4.2. Bibliografie

Bibliografie se vytvářejí pomocí prostředí `thebibliography`. Každá položka začíná

```
\bibitem[label]{marker}
```

Řetězcem `marker` se potom ve svém dokumentu odkazujete na danou knihu, článek nebo referát:

```
\cite{marker}
```

Pokud nepoužijete nepovinný argument `label`, \LaTeX položkám přidělí jednoznačná čísla. Parametr příkazu `\begin{thebibliography}` určuje, kolik místa se má vyhradit pro číselné labely. V následujícím příkladě hodnotou `{99}` \LaTeX u říkáme, že větší číslo labelu než 99 se nevyskytne.

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

Partl [1] navrhl aby...

Literatura

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

U větších dokumentů stojí za úvahu použití programu `BibTeX`, který je součástí většiny \TeX ových distribucí. Tento program umožňuje udržovat bibliografickou databázi a získávat z ní informace o jejích položkách, na které jste se ve svém dokumentu odkázali. Vizualní podoba bibliografií vygenerovaných programem `BibTeX` je postavena na konceptu stylů, takže si můžete vybrat, v kterém z mnoha rozložení chcete svou bibliografii vysázet.

Novinkou při přípravě bibliografie jsou nástroje `biblatex` a `biber`.

4.3. Indexování

Velmi užitečnou pomůckou pro orientaci v knize může být index (též se říká rejstřík). Ten lze v \LaTeX u (ve spolupráci s programem `makeindex`⁵ či `xindy`) vytvořit snadno. Tento úvod vysvětlí základní příkazy pro generování indexu. Detailnější pohled poskytuje *The \LaTeX Companion* [3].

Abychom v \LaTeX u mohli vytvářet indexy, je potřeba nahrát balík `makeidx` v preambuli pomocí:

```
\usepackage{makeidx}
```

Speciální indexovací příkazy zpřístupníme umístěním

```
\makeindex
```

do preambule vstupního souboru.

Položku indexu specifikujeme pomocí příkazu

```
\index{key@formatted_entry}
```

formatted_entry bude zobrazeno v indexu a *key* se použije pro třídění. *formatted_entry* je nepovinné a není-li uvedeno, použije se *key*. Příkaz `\index` uvedete na těch místech dokumentu, kam chcete, aby index ukazoval. Tabulka 4.2 vysvětluje syntax.

Tabulka 4.2: Příklady syntaxe klíčů indexů.

Příklad	Položka indexu	Komentář
<code>\index{hello}</code>	hello, 1	Obyčejná položka
<code>\index{hello!Peter}</code>	Peter, 3	Podpoložka ‘hello’
<code>\index{Sam@\textsl{Sam}}</code>	<i>Sam</i> , 2	Formátovaná položka
<code>\index{Lin@\textbf{Lin}}</code>	Lin , 7	Formátovaná položka
<code>\index{Kaese@K"ase}</code>	Käse , 33	Formátovaná položka
<code>\index{ecole@'ecole}</code>	école, 4	Formátovaná položka
<code>\index{Jenny textbf}</code>	Jenny, 3	Formátované číslo strany
<code>\index{Joe textit}</code>	Joe, 5	Formátované číslo strany

Při zpracování vstupního souboru \LaTeX ke každému výskytu `\index` zapíše do pomocného souboru data popisující příslušnou indexovou položku (včetně čísla strany, na které se nachází). Tento pomocný soubor má stejné

⁵Na systémech, které nepodporují jména souborů delší než osm znaků, se program může jmenovat `makeidx`.

jméno, jako vstupní soubor, ale jinou příponu (`.idx`). Tento pomocný soubor lze pak zpracovat programem `makeindex`.

```
makeindex filename
```

Program `makeindex` z indexových položek vygeneruje setříděný index. Tento výsledný index se uloží do souboru s příponou `.ind`. Při následujícím zpracování vstupního souboru \LaTeX em je soubor `.ind` vložen do dokumentu do místa, kde je uvedeno

```
\printindex
```

Balík `showidx` dostupný v \LaTeX 2_ε vypíše do levého okraje dokumentu indexové položky uvedené na příslušném řádku. To je užitečné pro kontrolní čtení dokumentu a kontrolu vygenerovaného indexu.

Pokud je použit neopatrně, příkaz `\index` může ovlivnit vzhled vašeho dokumentu.

Mé slovo `\index{slovo}`. Na rozdíl od `slovo\index{slovo}`. Všimněte si pozice tečky.

Mé slovo `.` Na rozdíl od `slovo.` Všimněte si pozice tečky.

4.4. Efektní záhlaví

Balík `fancyhdr`⁶ autora Piet van Oostruma, poskytuje několik jednoduchých příkazů, pomocí kterých můžete přizpůsobit vzhled záhlaví a pat stránek svého dokumentu. Aplikací tohoto balíku je např. záhlaví této stránky.

Problémem při přizpůsobování záhlaví a pat je, jak do nich umístit název aktuální sekce, kapitoly a podobných věcí. \LaTeX to provádí ve dvou krocích. V definici záhlaví a paty můžete použít příkazy `\rightmark` a `\leftmark`, které reprezentují název aktuální sekce, resp. kapitoly. Hodnoty těchto dvou příkazů jsou přepsány, kdykoliv se zpracovává `\chapter` nebo `\section`.

Kvůli maximální flexibilitě nepředefinováá příkaz `\chapter` (a podobné) hodnotu `\rightmark` a `\leftmark` přímo. Místo toho `\chapter` volá jiný příkaz (`\chaptermark`, `\sectionmark` nebo `\subsectionmark`), který je za předefinování `\rightmark` a `\leftmark` zodpovědný.

Chcete-li změnit, jak je jméno kapitoly zobrazeno v záhlaví, stačí, když pomocí „`renew`“ změníte definici příkazu `\chaptermark`.

Obrázek 4.1 ukazuje možné nastavení balíku `fancyhdr`, které zobrazuje záhlaví zhruba tak, jak je použito v tomto dokumentu. Doporučujeme obstarat si na adrese uvedené v poznámce pod čarou na začátku sekce dokumentaci k tomuto balíku.

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

```

\documentclass{book}
\usepackage{fancyhdr} % Načteme si balíček fancyhdr.
\pagestyle{fancy} % Aktivujeme na záhlaví a zápatí styl fancy.
% Následujícími příkazy si zajistíme, že nadpisy
% kapitol a sekcí budou malými písmeny:
\renewcommand{\chaptermark}[1]{%
    \markboth{#1}{}}
\renewcommand{\sectionmark}[1]{%
    \markright{\thesection\ #1}}
% Vymaž vše ze současného záhlaví a zápatí:
\fancyhf{}
% Nastav objekty v záhlaví:
\fancyhead[LE,RO]{\bfseries\thepage}
\fancyhead[LO]{\bfseries\rightmark}
\fancyhead[RE]{\bfseries\leftmark}
\renewcommand{\headrulewidth}{0.5pt} % Síla linky v záhlaví.
\renewcommand{\footrulewidth}{0pt} % Síla linky v zápatí.
\addtolength{\headheight}{0.5pt} % Navýšení výšky záhlaví.
% Připravme si staronový styl plain:
\fancypagestyle{plain}{%
    \fancyhead{} % Vymaž vše ze záhlaví.
    \renewcommand{\headrulewidth}{0pt} % Odstraň linku v záhlaví.
} % Konec stylu fancyplain.

```

Obrázek 4.1: Ukázka nastavení fancyhdr.

4.5. Balík Verbatim

Už dříve jsme se seznámili s *prostředím verbatim*. V této sekci představíme *balík verbatim*. Tento balík v podstatě vznikl přepsáním prostředí *verbatim*, odstraněním některých omezení tohoto původního prostředí a přidáním nové funkcionality. Balík *verbatim* poskytuje příkaz

```
\verbatiminput{filename}
```

který umožňuje vložit do dokumentu ASCII text, jako kdyby tento text byl uveden uvnitř prostředí *verbatim*.

Jelikož je balík *verbatim* součástí kolekce *tools*, měl by být na většině systémů předinstalován. Více informací naleznete v [10].

4.6. Instalace dodatečných balíčků

Součástí většiny L^AT_EXových instalací je velké množství předinstalovaných stylových balíčků a ještě mnohem více je jich dostupných přes Internet. Základní místo kde hledat stylové balíky je CTAN (<http://www.ctan.org/>).

Balíky jako např. *geometry* nebo *hyphenat* se obvykle skládají ze dvou souborů: jednoho s příponou *.ins* a druhého s příponou *.dtx*. Často je přiložen i soubor *readme.txt*, v kterém najdete stručný popis daného balíku a který byste si samozřejmě měli nejdřív ze všeho přečíst.

Jakmile si soubory příslušného balíku nakopírujete na svůj počítač, je potřeba je ještě zpracovat, abyste (a) své T_EXové distribuci o příslušném stylovém balíku řekli a (b) získali jeho dokumentaci. První část provedete takto:

1. Spusťte L^AT_EX na soubor *.ins*. Tím vznikne soubor *.sty*.
2. Přemístěte *.sty* soubor do místa, kde ho vaše distribuce bude schopná najít, což je obvykle adresář `.../localtexmf/tex/latex` (uživatelé Windows a OS/2 samozřejmě používají obrácená lomítka).
3. Obnovte databázi jmen souborů své distribuce. Pro různé L^AT_EXové distribuce je potřeba spustit různé příkazy: `teTEX`, `fpTEX - texhash`; T_EX Live, `web2c - mktexlsr`; MiK_TE_X - `initexmf -update-fndb` nebo použijte grafické rozhraní.

Druhou část – získání dokumentace z *.dtx* souboru – provedete takto:

1. Spusťte L^AT_EX na soubor *.dtx*. Tím vznikne soubor *.dvi*. Je možné, že bude potřeba spustit L^AT_EX několikrát, aby ve vygenerovaném dokumentu správně sestavil křížové odkazy.

2. Zkontrolujte, že mezi soubory, které L^AT_EX vygeneroval, je i soubor `.idx`. Jestliže tento soubor vygenerován nebyl, pokračujte krokem 5.
3. Index vygenerujete pomocí :

`makeindex -s gind.ist name`

 (kde *name* nahradíte jménem hlavního souboru bez přípony).
4. Znovu spusťte L^AT_EX na soubor `.dtx`.
5. Pro pohodlnější čtení si můžete vygenerovat soubor `.ps` nebo `.pdf`.

Někdy L^AT_EX vygeneruje i soubor `.glo` (glossary – vysvětlivky). V tom případě spusťte následující příkaz mezi krokem 4 a 5:

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

Spusťte L^AT_EX na soubor `.dtx` ještě jednou a potom pokračujte krokem 5.

4.7. Pracujeme s pdfL^AT_EXem

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

PDF je formát hypertextových dokumentů. Podobně jako u webových stránek mohou být části textu označeny jako odkazy. Tyto odkazy ukazují na jiná místa v aktuálním nebo jiném dokumentu. Kliknutím na odkaz se přenesete na příslušné místo. V kontextu L^AT_EXu to znamená, že všechny výskyty `\ref` a `\pageref` se stávají odkazy. Kolekcemi odkazů se navíc stávají obsah, index a všechny ostatní podobné struktury.

Většina dnešních webových stránek je napsána v HTML (*HyperText Markup Language*). Pokud jde o psaní vědeckých dokumentů, má tento formát dvě významné nevýhody:

1. Vkládání matematických vzorců do HTML dokumentů obecně není podporováno. Tato podpora je sice standardizována, ale většina dnešních prohlížečů tento standard nepodporuje nebo jim chybí potřebné fonty.
2. Tisk HTML dokumentů je sice možný, ale výsledky se na různých platformách a s použitím různých prohlížečů značně liší. Kvalita je na míle vzdálená té, na kterou jsme zvyklí v L^AT_EXovém světě.

Mnoho lidí se pokusilo vytvořit překladač z L^AT_EXu do HTML. Některé z těchto pokusů byly dokonce docela úspěšné – v tom smyslu, že příslušné překladače jsou schopné z L^AT_EXových vstupních souborů vytvořit webové stránky, které lze přečíst... Jejich výsledky by ale rozhodně nešlo označit jako uspokojivý. Jakmile navíc začnete s převody dokumentů obsahujících složitější L^AT_EXové rysy a externí balíky, výsledky se podstatně zhorší. Autoři, kteří chtějí zachovat unikátní typografickou kvalitu svých dokumentů i na webu, se proto orientují na PDF (*Portable Document Format*), který

na jedné straně zachovává rozložení dokumentu, na straně druhé umožňuje používat hypertextovou navigaci. Většina prohlížečů je dnes vybavena pluginy, které umožňují přímé prohlížení PDF dokumentů.

I když na téměř každé platformě existují prohlížeče souborů ve formátu DVI a PS, prohlížení PDF dokumentů pomocí programů Acrobat Reader a Xpdf je mnohem oblíbenější. Výrobou PDF verzí svých dokumentů je tedy učiníte mnohem přístupnějšími svým potenciálním čtenářům.

4.7.1. PDF dokumenty pro WWW

Vytvoření PDF souboru z L^AT_EXových zdrojů je velmi jednoduché a to díky programu pdfT_EX vytvořenému Hàn Thê Thànhem. Tam, kde T_EX vytvoří DVI soubor, pdfT_EX vytvoří PDF soubor. Existuje také pdfL^AT_EX, který vytvoří PDF z L^AT_EXových zdrojů.

Formáty pdfT_EX i pdfL^AT_EX jsou většinou moderních T_EXových distribucí (např. teT_EX, fpT_EX, MiK_T_EX, T_EX Live and CMacT_EX) nainstalovány automaticky.

K vytvoření výstupu v PDF místo v DVI stačí nahradit příkaz `latex file.tex` příkazem `pdflatex file.tex`. Na systémech, kde se L^AT_EX nepouští z příkazové řádky, bývá speciální tlačítko v T_EXControlCenter.

Stejně jako v L^AT_EXu lze i v pdfL^AT_EXu nastavit velikost papíru pomocí nepovinného argumentu příkazu `\documentclass`, např. `a4paper` nebo `letterpaper`. PdfT_EX navíc potřebuje znát velikost papíru, aby dokázal určit fyzickou velikost stránek v PDF souboru. Pokud použijete balík `hyperref` (viz stránka 92), velikost papíru bude určena automaticky. Jinak musíte velikost určit sami přidáním následujících řádek do preambule svého dokumentu:

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

Následující sekce podrobněji rozebere detaily týkající se rozdílů mezi normálním L^AT_EXem a pdfL^AT_EXem. Hlavní rozdíly jsou ve třech oblastech: fontech, formátu vkládaných obrázků a ručním konfigurování hypertextových odkazů.

4.7.2. Fonty

pdfL^AT_EX umí pracovat se všemi druhy fontů (PK bitmapy, TrueType, POSTSCRIPT Type 1...), ale implicitní formát fontů v L^AT_EXu, PK bitmapy, poskytuje při zobrazení dokumentu v Acrobat Readeru velmi špatné výsledky. Nejlepšího vzhledu dosáhneme používáním fontů POSTSCRIPT Type 1.

Moderní T_EXové instalace jsou proto nakonfigurovány tak, aby fonty Type 1 používaly automaticky. Jestliže je to i případ vaší instalace, můžete tuto sekci přeskočit.

POSTSCRIPT Type 1 implementace Computer Modern fontů a AMS-Fonts byla vytvořena společností Blue Sky Research and Y&Y, Inc., která převedla autorská práva na Americkou matematickou společnost. Fonty byly zpřístupněny počátkem roku 1997 a dnes jsou součástí většiny T_EXových distribucí.

Pokud ale pomocí L^AT_EXu vytváříte dokumenty v jazycích jiných, než je angličtina, mohly by se vám lépe hodit fonty EC, LH, nebo CB (viz diskuze o fontech OT1 na straně 24). Vladimír Volovich vytvořil skupinu fontů cm-super, která pokrývá celou množinu fontů EC/TC, EC Concrete, EC Bright a LH. Je dostupná na [CTAN://tex-archive/fonts/ps-type1/cm-super](http://ctan://tex-archive/fonts/ps-type1/cm-super) a také je součástí distribucí T_EXLive 7 a MiK_TE_X.

Podobné fonty Type 1 CB Greek vytvořené Apostolem Syropoulossem jsou dostupné na [CTAN://tex-archive/fonts/greek/cb](http://ctan://tex-archive/fonts/greek/cb). Bohužel ani jedna z těchto dvou sad nedosahuje kvality Type1 CM fontů od Blue Sky/Y&Y. Na rozdíl od fontů Blue Sky je totiž u fontů těchto dvou sad použito automatických hintů, které dávají v nízkém rozlišení na obrazovce horší výsledek. Na tiskárně s vysokým rozlišením jsou ale výsledky srovnatelné.

Pokud připravujete dokumenty v jednom z jazyků používajících latin-skou abecedu, máte několik dalších možností:

- Můžete použít balík *aeguill* (*Almost European Computer Modern with Guillemets*). Do preamble dokumentu stačí vložit řádku.
`\usepackage{aeguill}`
 a místo fontů EC se použijí virtuální fonty AE.
- Nebo můžete použít balík *mltex*. Ale ten bude fungovat jen pokud vámi používaný pdf_TE_X byl zkompileován s přepínačem *mltex*.

Virtuální font AE před T_EXem „předstírá“, že obsahuje (a tedy dává T_EXu k dispozici) všech 256 znaků. Většinu chybějících znaků přebírá z CM fontů, přičemž mění jejich pořadí tak, aby odpovídalo pořadí používaném EC. Tímto způsobem lze používat skvělé CM fonty ve formátu Type 1, které jsou na většině systémů dostupné. Jelikož font je nyní v kódování T1, dělení slov pro latinkové jazyky bude fungovat správně. Jedinou nevýhodou je, že v Acrobat Readeru nelze umělé AE znaky vyhledávat, takže akcentované znaky v dokumentu pomocí příkazu `Find` nenajdete.

Podobným řešením pro ruštinu je použití virtuálních fontů C1, dostupných na [ftp://ftp.vsu.ru/pub/tex/font-packs/c1fonts](http://ftp.vsu.ru/pub/tex/font-packs/c1fonts). Fonty kombinují standardní Type 1 fonty CM z kolekce Bluesky a Type 1 fonty CMCYR z kolekce Paradissa a BaKoMa, které jsou všechny dostupné z CTAN. Fonty Paradissa (a potažmo C1 fonty) obsahují jen ty znaky Cyriliky, které se používají v ruštině.

Můžete také použít jiné POSTSCRIPT Type 1 fonty, např. některé z těch, které jsou součástí instalace Acrobat Readeru. Protože znaky těchto fontů mají jiné velikosti, rozložení textu stránek vašeho dokumentu bude jiné.

Obecně budou tyto fonty zabírat více místa než CM fonty, které jsou velmi úsporné. Dokument také nebude vypadat tak soudržně, protože Times, Helvetica a Courier (hlavní kandidáti pro použití) nebyly navrženy pro dobrou „spolupráci“ mezi sebou v jednom dokumentu.

K dispozici jsou dvě sady fontů: `pxfonts` (jejíž hlavním fontem pro text je *Palatino*) a balík `txfonts` (hlavní font *Times*). Abyste je mohli použít, stačí vložit následující řádky do preambule svého dokumentu:

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

Pozn.: Po zpracování dokumentu se v `.log` souboru objeví řádky typu

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

Tato varování říkájí, že font použitý v dokumentu nebyl nalezen. To je problém a musíte ho nějak vyřešit. Jinak totiž výsledný PDF dokument *nemusí vůbec zobrazit stránky s chybějícími znaky*.

Nedostupnost rodin EC fontů v kvalitě podobné CM fontům ve formátu Type 1 vedla k vytvoření fontů Latin Modern (LM), které jsou nyní k dispozici. Tyto fonty řeší problémy popsané výše. Je-li vaše T_EXová instalace aktuální, nejspíš tyto fonty obsahuje a vám stačí přidat

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

do preambule svého dokumentu a můžete s jejich pomocí vytvářet vynikající PDF výstup s plnou podporou všech znaků latinské abecedy.

4.7.3. Používání grafiky

Na vkládání grafiky do dokumentů je nejlepší balík `graphicx` (viz strana 81). Při použití tohoto balíku s pdfL^AT_EXem je potřeba v příkazu `\usepackage` uvést `pdftex` jako hodnotu nepovinného parametru *driver*:

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

V ukázce jsme také nahráli balík `color` (použití barev v dokumentech vystavených na webu je dnes běžné).

Použití grafiky v PdfL^AT_EXu má ale jeden problém: PdfL^AT_EX neumí zpracovat grafické soubory Encapsulated POSTSCRIPT. Pokud v příkazu `\includegraphics` neuvedete příponu souboru, `graphicx` si ji odvodí podle nastavení parametru *driver*. Pro `pdftex` lze použít jeden z formátů `.png`, `.pdf`, `.jpg` nebo `.mps` (METAPOST) – ale *ne* `.eps`.

Nejjednodušším řešením je zkonvertovat váš EPS soubor do formátu PDF pomocí utility `epstopdf`, která je dostupná na mnoha systémech. Tento

postup je vhodné použít pro vektorovou grafiku. Pro bitmapy (fotky, naskenované obrázky, atd.) je lepší použít formát PNG (ideální pro obrázky s málo barvami, např. u snímků obrazovky, angl. screenshot) nebo JPEG (ideální pro fotky, které ukládá velmi úsporně).

Na kreslení geometrických obrázků lze použít speciální programy, např. METAPOST, který je součástí většiny \TeX ových distribucí a je podrobně zdokumentován.

4.7.4. Hypertextové odkazy

Balík `hyperref` dokáže všechny reference v dokumentu přeměnit na hypertextové odkazy. Jediné, co musíte udělat, je přidat následující řádku *na konec* preamble svého dokumentu: `\usepackage[pdftex]{hyperref}`

Chování balíku `hyperref` je možno přizpůsobit pomocí mnoha nastavení:

- buď uvedenými jako seznam oddělený čárkami (za volbou `pdftex`)
`\usepackage[pdftex]{hyperref}`
- nebo na nových řádcích pomocí příkazu `\hypersetup{options}`.

Jediným povinným nastavením je `pdftex`. Pomocí ostatních, nepovinných, nastavení lze změnit implicitní chování.⁷ V následujícím seznamu jsou implicitní hodnoty vtištěny itálikou.

`bookmarks (=true, false)` při zobrazování dokumentu zobrazí, resp. skryje panel s bookmarky.

`unicode (=false, true)` umožňuje používat v bookmarcích Acrobat Readeru znaky nelatinkových jazyků.

`pdftoolbar (=true, false)` zobrazí, resp. skryje v AR panel nástrojů.

`pdfmenubar (=true, false)` zobrazí, resp. skryje v AR menu.

`pdffitwindow (=false, true)` přizpůsobí počáteční zvětšení prohlíženého PDF dokumentu.

`pdftitle (=text)` nastavuje titul, který je zobrazen v okně **Informace o dokumentu AR**.

`pdfauthor (=text)` nastavuje jméno autora daného dokumentu.

`pdfnewwindow (=false, true)` specifikuje, jestli se má otevřít nové okno ve chvíli, kdy následujeme odkazy mířící mimo aktuální dokument.

⁷Balík `hyperref` není omezen pouze na práci s pdf \TeX em. Lze ho nakonfigurovat tak, aby vkládal specifické PDF informace do DVI výstupu „normálního“ $\mathcal{E}\TeX$ u. DVI výstup je následně přesunut do souboru PS (pomocí `dvips`), odkud ho při konverzi PS do PDF umí získat Adobe Distiller.

`colorlinks` (`=false`, `true`) specifikuje, zda odkazy mají být obklopeny barevnými rámy (`false`) nebo zda mají být obarveny texty odkazů (`true`). Barva odkazů se dá zadat pomocí následujících nastavení (ukázány jsou i implicitní hodnoty):

```
linkcolor (=red) barva vnitřních odkazů (sekce, stránky, atd.),
citecolor (=green) barva odkazů na citace (bibliografie),
filecolor (=magenta) barva odkazů na soubory,
urlcolor (=cyan) barva URL odkazů (mail, web).
```

Pokud vám implicitní nastavení vyhovuje, použijte

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

Chcete-li mít seznam bookmarků otevřený a odkazy barevné (část parametru `=true` je nepovinná), použijte:

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

Barevným odkazům je lépe se vyhnout v případě, že plánujeme náš PDF dokument tisknout (barvy odkazů jsou po černobílém vytištění vidět jako odstíny šedé a špatně se čtou). Místo toho se můžou použít barevné rámečky, které vytištěné nejsou:

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

nebo nastavit černou barvu odkazů:

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

Když chcete jen zadat informaci pro Informaci o dokumentu, použijte:

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

Kromě automatického přidání hypertextových odkazů pro křížové odkazy je také možné začlenit odkazy explicitně pomocí

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

Například kód:

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

vyprodukuje výstup: The **CTAN** website.; kliknutím na název serveru přejdete na webovou stránku serveru CTAN.

Ukazuje-li odkaz na lokální soubor, použijte příkaz `\href`:

```
Celý dokument je uložen \href{manual.pdf}{zde.}
```

Tím vytisknete text „Celý dokument je uložen **zde.**“. Kliknutím na slovo „**zde.**“ otevřeme soubor `manual.pdf`. Jméno souboru se bere lokálně k aktuálnímu dokumentu.

Chce-li autor článku, aby mu čtenáři mohli rychle poslat email, může napsat příkaz `\href` do příkazu `\author` na titulní stránce dokumentu:

```
\author{Mary Oetiker <\href{mailto:mary@oetiker.ch}%  
{mary@oetiker.ch}>}$}
```

Kromě toho, že je nyní emailová adresa přítomna v PDF odkazu, je i vysázena na stránce, abychom o ni po vytištění dokumentu nepřišli.

4.7.5. Problémy s odkazy

Varování podobné tomuto:

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

se začnou objevovat v případě, že je číselný registr reinitializován, např. použitím příkazu `\mainmatter` z třídy dokumentů `book`. Ten reinitializuje registr s číslem stránky na jedničku. Ale protože i předmluva knihy má číslo stránky 1, žádný odkaz na „stranu 1“ už není unikátní. Proto ona hláška `duplicate ignored`, duplicitní registr byl ignorován.

Přidáním `plainpages=false` mezi nastavení `hyperref` vše napravíme. To ale bohužel pomůže jen s číselníky stran. Radikálnějším řešením je nastavení `hypertexnames=false`, ale to zase způsobí, že odkazy na stránky v indexu přestanou fungovat.

4.7.6. Problémy s bookmarky

Text zobrazený bookmarky vždy nevypadá tak, jak bychom čekali. Bookmark může zobrazit jen „obyčejný text“ a k dispozici máme tedy mnohem méně znaků než u textu dokumentu. `Hyperref` si obvykle takového problému všimne a upozorní na něj:

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

Řešením je uvést kromě „L^AT_EXového textu“, který nemusí jít použit v bookmarku, i „zjednodušený text“, který naopak v bookmarku použit půjde:

```
\texorpdfstring{TEX text}{Bookmark Text}
```

Matematické vzorce jsou typickými kandidáty pro problémy tohoto typu. Díky konstrukci:

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

bude v bookmarku místo „L^AT_EXového textu“ `\section{$E=mc^2$}` použit „zjednodušený text“ „E=mc²“.

Změny barev také nelze jednoduše promítnout do bookmarku:

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

vyprodukuje řetězec „redRed!“. Příkaz `\textcolor` je ignorován, ale jeho argument („red“) se vytiskne).

Použijete-li

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

výsledek bude mnohem lépe čitelný.

Pokud svůj dokument píšete v unicode, můžete použít nastavení `unicode` balíku `hyperref`, které umožňuje používat v bookmarku unicode znaky. To vám dává mnohem větší výběr znaků pro použití v `\texorpdfstring`.

Kompatibilita zdrojových kódů pro L^AT_EX a pdfL^AT_EX

Až na několik málo výjimek by váš L^AT_EXový zdrojový kód měl jít přeložit jak L^AT_EXem, tak pdfL^AT_EXem. Hlavní z těchto výjimek je vkládání grafiky. Jednoduchým řešením je při použití příkazu `\includegraphics` příponu souborů *neuvádět*. Při překladu pomocí „normálního“ L^AT_EXu se potom použije soubor s daným jménem a příponou `.eps`, zatímco při překladu pomocí pdfL^AT_EXu se použije jedna z přípon `.png`, `.pdf`, `.jpg` nebo `.mps` (v tomto pořadí). Je samozřejmě nutné, aby příslušný soubor byl přítomen v pracovním adresáři.

Chcete-li ve svém dokumentu použít jiný zdrojový kód podle toho, jestli se dokument překládá L^AT_EXem nebo pdfL^AT_EXem, můžete použít balík `ifpdf`⁸. Pokud tento balík ještě nemáte nainstalovaný, asi používáte MiK_TE_X, který ho ale doinstaluje automaticky sám, až se ho poprvé budete snažit použít. Tento balík definuje příkaz `\ifpdf`, pomocí kterého snadno napíšete obě verze kódu („DVI verzi“ i „PDF verzi“). V následujícím příkladu chceme,

⁸Více informací vysvětlujících proč použít tento balík je dostupných na stránce T_EX FAQ: <http://www.tex.ac.uk/cgi-bin/texfaq2html?label=ifpdf>

aby PostScriptová („normální“) verze byla černobílá (tuto verzi plánujeme tisknout a nechceme utrácet na barvě) a PDF verze (kterou budeme prohlížet jen na monitoru) barevná.

```

\RequirePackage{ifpdf} % Spouští se pdfTeX?
\ifpdf                % Ano, spouští.
  \documentclass[a4paper,12pt,pdftex]{book}
\else                  % Ne, není tomu tak.
  \documentclass[a4paper,12pt,dvips]{book}
\fi

\ifpdf % Používá-li se pdfTeX, zařaď Latin Modern.
  \usepackage{lmodern}
\fi
\usepackage[bookmarks, % Přidáme záložky...
            colorlinks,
            plainpages=false]{hyperref}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
\usepackage[english]{babel}
\usepackage{graphicx}
...

```

V příkladu nahráváme balík `hyperref` i při překladu „normální“ verze. To proto, aby příkaz `\href` fungoval i tam a nemuseli jsme všechna jeho použití uvádět podmíněně.

V dnešních distribucích (např. `TeX Live`) program `TeX` ve skutečnosti ukazuje na program `pdfTeX`. Rozlišení, jestli jako výstup produkovat DVI nebo PDF je učiněno na základě použitých nastavení třídy dokumentu. Kódem z předchozího příkladu získáte PDF výstup při použití `pdflatex` a DVI výstup při použití `latex`.

4.8. Vytváření prezentací

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

Výsledky své vědecké práce můžete prezentovat buď na tabuli, nebo pomocí připravené prezentace ze svého notebooku.

`pdfLaTeX` a třída `beamer` dohromady umožňují vytvářet PDF prezentace, které vypadají lépe než ty vytvořené pomocí PowerPointu a jednodušeji se používají, protože Acrobat Reader je dostupný na mnohem více platformách než PowerPoint.

Třída `beamer` používá balíky `graphicx`, `color` a `hyperref` s hodnotami nastavení vhodnými pro prohlížení dokumentu na monitoru.

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

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

\begin{document}

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

\section{Příklad}

\begin{frame}
  \frametitle{Co dělat v~neděli odpoledne}
  \begin{block}{Je možné\ldots}
    \begin{itemize}
      \item jít na vycházku se psem\pause
      \item číst knihu\pause
      \item škádlit kočku\pause
    \end{itemize}
  \end{block}
  a~mnoho dalšího\ldots
\end{frame}
\end{document}
```

Obrázek 4.2: Ukázkový kód použití třídy beamer.

Přeložením kódu z obrázku 4.2 pdfL^AT_EXem získáte PDF soubor s titulní stránkou následovanou stránkou obsahující několik položek, které budou zobrazovány po jedné během procházení prezentací.

Výhodou `beamer` oproti podobným balíkům je, že vytváří přímo PDF soubor (na rozdíl od balíku `prospcr`, pomocí kterého lze vytvořit PostScript soubor) a nevyžaduje následné úpravy (na rozdíl od balíku `ppower4`).

Pomocí třídy `beamer` můžete ze stejného zdrojového souboru vytvořit několik verzí (módů) svého dokumentu. Zdrojový soubor může obsahovat speciální instrukce (v lomených závorkách) pro jednotlivé módy. Toto jsou dostupné módy:

beamer pro prezentace (o kterých jsme mluvili dosud),

trans pro slajdy,

handout pro verzi pro tisk.

Implicitním módem je `beamer`. Změnu lze udělat globálním nastavením jiného módu, např. `\documentclass[10pt,handout]{beamer}`, pro tvorbu doplňujících podkladů řečníkovi k prezentaci.

Vzhled prezentace na monitoru závisí na tématu, které si zvolíte. Na výběr máte témata dodávaná ve třídě `beamer`, nebo si můžete téma sami vytvořit. Více informací najdete v souboru `beameruserguide.pdf` – obsáhlé dokumentaci třídy `beamer`.

Podrobněji se teď podívejme na kód na obrázku 4.2.

Pro obrazkovou verzi prezentace (`\mode<beamer>`) jsme si zvolili téma *Goettingen*, které má navigační panel zintegrován do obsahu. Nastavit lze velikost panelu (v našem případě 22 mm) a jeho pozici (zde na pravé straně textu). Nastavení `hideothersubsections` ukazuje jména kapitol a podsekce aktuální kapitoly. Módy `\mode<trans>` a `\mode<handout>` nemají žádné speciální nastavení a používají tedy standardní vzhled.

Příkazy `\title`, `\author`, `\institute` a `\titlegraphic` o jednom povinném parametru nastavují obsah titulní stránky. Nepovinné parametry příkazů `\title[]{}{}` a `\author[]{}{}` umožňují zadat speciální verzi titulku a jména autora pro zobrazení na panelu tématu *Goettingen*.

Tituly a podtituly jsou v panelu vytvořeny pomocí normálních příkazů `\section` a `\subsection`, které uvedete *mimo* prostředí `frame`.

Málo navigační ikony na spodu obrazovky umožňují navigovat dokument. Jejich přítomnost nezávisí na vybraném tématu.

Obsah každého slajdu (resp. obrazovky) musíme umístit do prostředí `frame`. Nepovinný parameter v lomených závorkách (`<` a `>`) může zamezit tomu, aby konkrétní `frame` byl v jedné z verzí použit. V našem příkladu by díky `<handout:0>` v letákové verzi nebyla použita první stránka.

Důrazně se doporučuje uvádět titul každého slajdu (kromě titulního slajdu) – pomocí příkazu `\frametitle{}`. Pokud potřebujete i podtitul,

použijte prostředí `block`, jak je vidět v příkladu. Všimněte si, že příkazy pro sekce (`\section`, `\subsection`) nevytvářejí žádný výstup na vlastním slajdu.

Příkaz `\pause` v prostředí `itemize` umožňuje odhalovat položky postupně.

Další prezentační efekty lze dosáhnout příkazy `\only`, `\uncover`, `\alt` a `\temporal`. V mnoha místech lze použít ostré závorky pro další přizpůsobení prezentace.

Můžeme doporučit projít a přečíst si `beameruserguide.pdf` – dokumentaci k třídě `beamer` – abyste získali úplný obrázek o možnostech této třídy. Na vývoji Beameru se stále pokračuje, nejnovější informace můžete získat na domovské webové stránce:

<http://latex-beamer.sourceforge.net/>

Kapitola 5.

Tvorba matematické grafiky

Podobně jako zadáváme text, můžeme \LaTeX u zadat pokyny pro vytvoření grafického výstupu. Možnosti, které při tom máme, jsou trochu omezené, ale existuje řada \LaTeX ových rozšíření, které tato omezení překonávají. V této sekci se o několika z nich dozvíte.

5.1. Úvodní přehled

Prostředí `picture` umožňuje programování obrázků přímo prostřednictvím \LaTeX u. Podrobný popis lze nalézt v *\LaTeX Manual* [1]. Existují výrazná omezení, protože sklony úseček stejně jako poloměry kružnic jsou omezené na malou skupinu hodnot. Na druhou stranu ale $\text{\LaTeX} 2_{\epsilon}$ prostředí `picture` přináší příkaz `\qBezier` („q“ jako quadratic – kvadratické). Kvadratickými Bézierovými křivkami lze uspokojivě aproximovat mnoho často používaných křivek (např. kružnice, elipsy nebo řetězovky), i když to může vyžadovat trochu matematické dřiny. Jestliže se navíc pro generování `\qBezier` bloků \LaTeX ových vstupních souborů použije programovací jazyk, např. typu Java, prostředí `picture` se stává docela mocným.

Přestože programování obrázků přímo v \LaTeX u je výrazně omezené a často docela únavné, má své výhody. Dokumenty vytvořené tímto způsobem zabírají méně místa a není třeba udržovat žádné grafické soubory.

Balíky, např. `epic` a `eepic` (popsané např. v *The \LaTeX Companion* [3]) nebo `pstricks` pomáhají eliminovat omezení původního prostředí `picture` a výrazně posilují grafické schopnosti \LaTeX u.

Zatímco první dva balíky jen obohacují prostředí `picture`, balík `pstricks` má své vlastní kreslicí prostředí, `pspicture`. Síla `pstricks` vychází z toho, že tento balík intenzivně využívá možnosti jazyka POSTSCRIPT. Mnoho balíků bylo navíc vytvořeno pro konkrétní použití. Řada z těchto balíků je detailně popsána v *The \LaTeX Graphics Companion* [4], neplést si prosím s *The \LaTeX Companion* [3].

Asi nejsilnějším grafickým nástrojem spřízněným s \LaTeX em je METAPOST , dvojče programu METAFONT Donalda E. Knutha. METAPOST obsahuje sofistikovaný programovací jazyk METAFONTu . Na rozdíl od METAFONTu (který generuje bitmapy) generuje METAPOST zapouzdřené POSTSCRIPT soubory, které mohou být importovány do \LaTeXu . Úvod do programu METAPOST naleznete v *A User's Manual for METAPOST* [15] nebo v tutoriálu [17].

Velmi důkladná diskuze \LaTeX ových a TEX ových strategií pro zacházení s grafikou (a fonty) je součástí *TEX Unbound* [16].

5.2. Prostředí picture

autor: Urs Oswald <osurs@bluewin.ch>

5.2.1. Základní příkazy

Prostředí `picture`¹ se vytvoří jedním z následujících dvou příkazů

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

nebo

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

Čísla x , y , x_0 , y_0 jsou odkazy na `\unitlength`, které mohou být kdykoliv nastaveny na původní hodnoty (ale ne uvnitř prostředí `picture`) např. příkazem

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

Implicitní hodnota `\unitlength` je `1pt`. První pár, (x, y) , zajistí v rámci dokumentu vynechání obdélníkového místa pro obrázek. Volitelný druhý pár, (x_0, y_0) , přiřadí libovolné souřadnice spodnímu levému rohu vyhrazeného obdélníku.

¹Věřte nevěřte, ale prostředí `picture` je k dispozici přímo ve standardním $\text{\LaTeX} 2_{\epsilon}$, není tedy třeba nahrávat žádné speciální balíky.

Většina kreslicích příkazů má jednu ze dvou forem

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

nebo

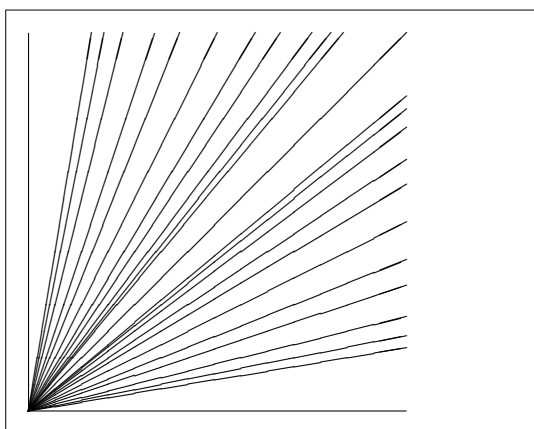
```
\multiput(x,y)(\Delta x,\Delta y){n}{object}
```

Výjimkou jsou Bézierovy křivky, které se kreslí příkazem

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

5.2.2. Řádkové segmenty

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



Řádkové segmenty se kreslí příkazem

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

Příkaz `\line` má dva argumenty:

1. směrový vektor,
2. délka (*length*).

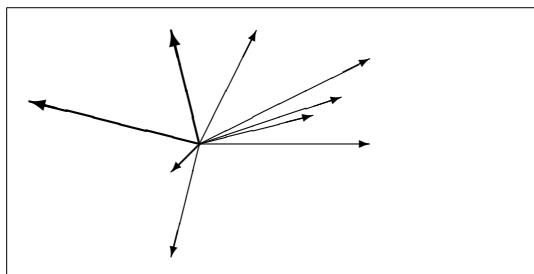
Komponenty směrového vektoru jsou omezeny na celá čísla

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

a musejí být nesoudělné (nemající kromě jedničky žádného společného dělitele). Obrázek ukazuje všech dvacet pět možných směrových hodnot v prvním kvadrantu. Argument délka je relativní vzhledem k `\unitlength`. Délka je vertikální souřadnice v případě vertikálního řádkového segmentu a horizontální souřadnice ve všech ostatních případech.

5.2.3. Šipky

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



Šipky se kreslí příkazem

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

Hodnoty směrových vektorů šipek jsou omezeny ještě více než u řádkových segmentů, konkrétně se jedná o celá čísla

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

Opět platí, že komponenty musí být nesoudělné. Všimněte si efektu příkazu `\thicklines` na šipky ukazující nahoru doleva.

5.2.4. Kružnice

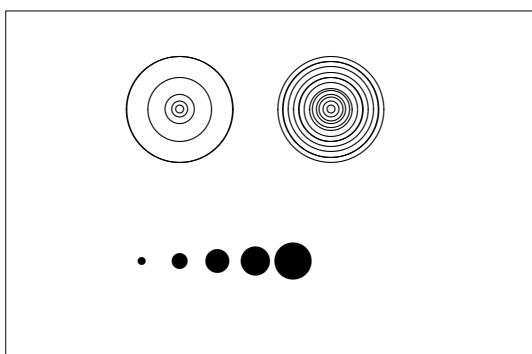
```

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

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

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

```



Příkaz

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

nakreslí kružnici se středem (x, y) a průměrem (ano, nikoliv poloměrem) $diameter$. Maximální průměr, který prostředí `picture` akceptuje, je zhruba 14 mm, ale použít nelze ani některé menší průměry. Příkaz `\circle*` nakreslí disky (vyplněné kružnice).

Podobně jako u řádkových segmentů je možno použít přídatné balíky, např. `eepic` nebo `pstricks`. Tyto balíky jsou důkladně popsány v *The L^AT_EX Graphics Companion* [4].

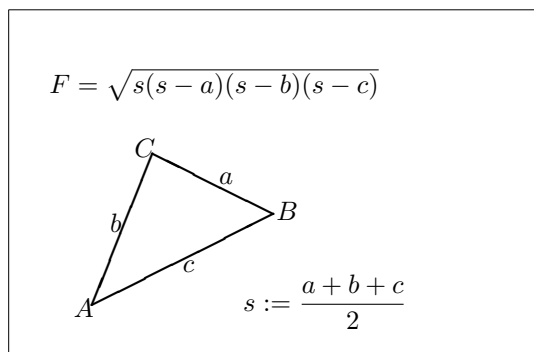
Kdo se nebojí provádět (sám nebo pomocí programu) nezbytné výpočty, může v prostředí `picture` z kvadratických Béziových křivek složit dohromady libovolné kružnice a elipsy. Příklady a zdrojové soubory v Javě jsou uvedeny v *Graphics in L^AT_EX 2_ε* [17].

5.2.5. Text a vzorce

```

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

```



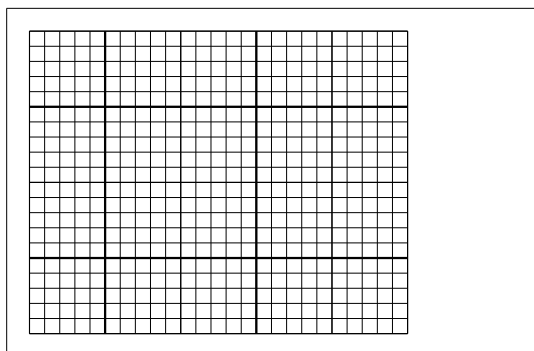
Jak ukazuje tento příklad, text a vzorce mohou být obvyklým způsobem vepsány do prostředí `picture` pomocí příkazu `\put`.

5.2.6. `\multiput` a `\linethickness`

```

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

```



Příkaz

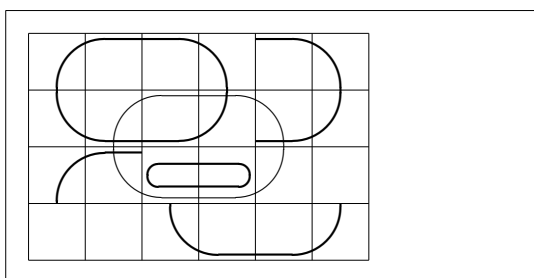
```
\multiput(x,y)(\Delta x,\Delta y){n}{object}
```

má 4 argumenty: počáteční bod, vektor pro překlad z jednoho objektu do jiného, počet objektů a objekt, který se má nakreslit.

Příkaz `\linethickness` se vztahuje na horizontální a vertikální linkové segmenty, ale ne na šikmé linkové segmenty ani kružnice. `\linethickness` se také vztahuje na kvadratické Bézierovy křivky!

5.2.7. Ovály

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



Příkaz

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

nebo

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

nakreslí ovál se středem (x, y) , šířkou w a výškou h . Volitelné *poziční* argumenty `b`, `t`, `l`, `r` se vztahují k „horní“, „spodní“, „levé“ a „pravé“ straně a mohou být kombinovány, jak je ukázáno na obrázku.

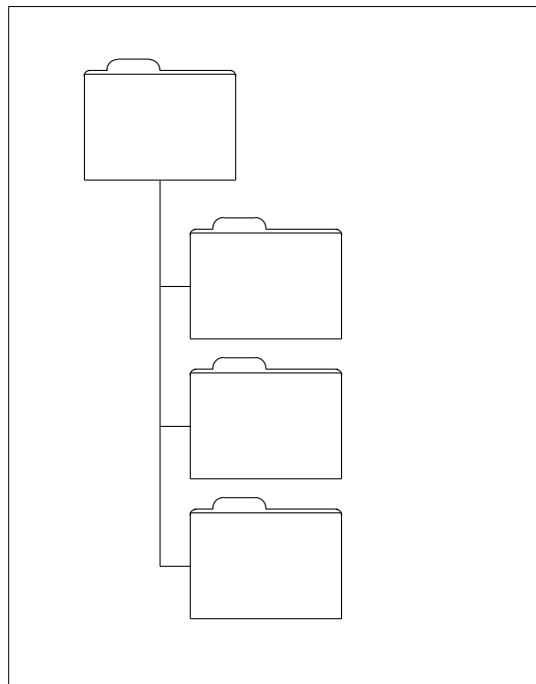
Šířku čar je možno upravovat buď příkazem `\linethickness{length}` nebo příkazy `\thinlines` a `\thicklines`. `\linethickness{length}` se vztahuje jen na horizontální a vertikální čáry (a kvadratické Bézierovy křivky), zatímco `\thinlines` a `\thicklines` lze navíc použít na šikmé čárové segmenty, kružnice a ovály.

5.2.8. Vícenásobné použití předdefinovaných boxů s obrázky

```

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

```



Box s obrázkem je možno *deklarovat* příkazem

```
\newsavebox{name}
```

a následně *definovat* pomocí

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

a potom *vykreslit* libovolně mnohokrát pomocí

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

Nepovinný parametr *position* specifikuje *referenční bod* saveboxu. V našem příkladu je tento parameter specifikován hodnotou *bl*, což umístí referenční bod do spodního levého rohu saveboxu (*bottom* – spodní část a *left* –

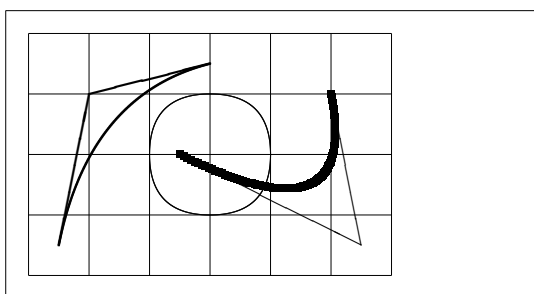
– vlevo). Zbylé hodnoty, ze kterých je možno složit specifikaci pozice, jsou **t** (top – horní část) a **r** (right – vpravo).

Argument *name* se odkazuje na L^AT_EX 2_εové „skladiště materiálu“ a má proto charakter příkazu, proto jsou v našem příkladu zpětná lomítka. Boxy nesoucí obrázky mohou být vnořené: v našem příkladě je použito `\foldera` uvnitř definice `\folderb`.

Museli jsme použít příkaz `\oval`, protože příkaz `\line` nefunguje, pokud je délka segmentu menší než zhruba 3 mm.

5.2.9. Kvadratické Bézierovy křivky

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



Jak tento příklad ukazuje, rozdělení kružnice na čtyři kvadratické Bézierovy křivky nevede k uspokojivému výsledku. Křivek je potřeba alespoň osm. Obrázek opět ukazuje efekt příkazu `\linethickness` na horizontální a vertikální čáry a příkazů `\thinlines` a `\thicklines` na šikmé čárové segmenty. Ukazuje také, že oba typy příkazů ovlivňují kvadratické Bézierovy křivky, každý z příkazů „přepisující“ nastavení toho předchozího.

Nechť $P_1 = (x_1, y_1)$, $P_2 = (x_2, y_2)$ značí koncové body a m_1, m_2 příslušné sklony kvadratické Bézierovy křivky. Střední kontrolní bod $S = (x, y)$ potom splňuje rovnici

$$\begin{cases} x = \frac{m_2 x_2 - m_1 x_1 - (y_2 - y_1)}{m_2 - m_1}, \\ y = y_i + m_i(x - x_i) \quad (i = 1, 2). \end{cases} \quad (5.1)$$

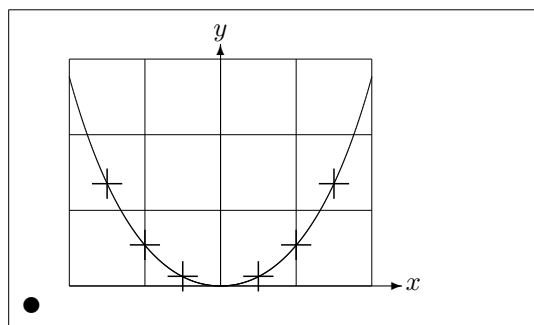
Graphics in L^AT_EX 2_ε [17] obsahuje Javový program, který generuje nezbytnou příkazovou řádku pro `\qBezier`.

5.2.10. Řetězovka

```

\setlength{\unitlength}{1cm}
\begin{picture}%
  (4.3,3.6)(-2.5,-0.25)
\put(-2,0){\vector(1,0){4.4}}
\put(2.45,-.05){$x$}
\put(0,0){\vector(0,1){3.2}}
\put(0,3.35){\makebox(0,0){$y$}}
\qBezier(0.0,0.0)(1.2384,0.0)
  (2.0,2.7622)
\qBezier(0.0,0.0)(-1.2384,0.0)
  (-2.0,2.7622)
\linethickness{.075mm}
\multiput(-2,0)(1,0){5}
  {\line(0,1){3}}
\multiput(-2,0)(0,1){4}
  {\line(1,0){4}}
\linethickness{.2mm}
\put(.3,.12763){\line(1,0){.4}}
\put(.5,-.07237){\line(0,1){.4}}
\put(-.7,.12763){\line(1,0){.4}}
\put(-.5,-.07237){\line(0,1){.4}}
\put(.8,.54308){\line(1,0){.4}}
\put(1,.34308){\line(0,1){.4}}
\put(-1.2,.54308){\line(1,0){.4}}
\put(-1,.34308){\line(0,1){.4}}
\put(1.3,1.35241){\line(1,0){.4}}
\put(1.5,1.15241){\line(0,1){.4}}
\put(-1.7,1.35241){\line(1,0){.4}}
\put(-1.5,1.15241){\line(0,1){.4}}
\put(-2.5,-0.25){\circle*{0.2}}
\end{picture}

```



Každá ze symetrických polovin paraboly $y = \cosh x - 1$ na tomto obrázku lze přibližně získat pomocí Bézierovy křivky. Pravá polovina křivky končí v bodě $(2, 2.7622)$, kde má sklon hodnotu $m = 3.6269$. Pokud opět použijeme rovnici (5.1), můžeme vypočítat střední kontrolní body, které v našem případě jsou $(1.2384, 0)$ a $(-1.2384, 0)$. Křížky značí body *skutečné* paraboly. Chyba, menší než jedno procento, je stěží postřehnutelná.

V následujícím příkladu je u `\begin{picture}` použit nepovinný parametr. Obrázek se definuje pomocí vhodných „matematických“ souřadnic,

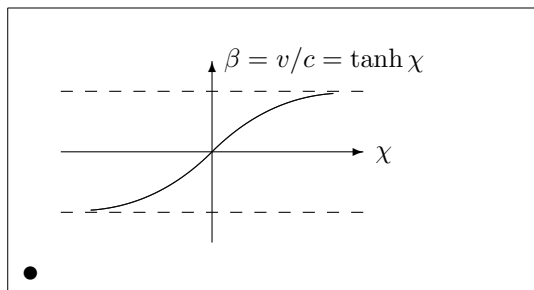
zatímco příkazem

```
\begin{picture}(4.3,3.6)(-2.5,-0.25)
```

přičadíme levému dolnímu rohu (viz černý kotouč) souřadnice $(-2.5, -0.25)$.

5.2.11. Rychlost ve speciální teorii relativity

```
\setlength{\unitlength}{0.8cm}
\begin{picture}(6,4)(-3,-2)
  \put(-2.5,0){\vector(1,0){5}}
  \put(2.7,-0.1){$\chi$}
  \put(0,-1.5){\vector(0,1){3}}
  \multiput(-2.5,1)(0.4,0){13}
    {\line(1,0){0.2}}
  \multiput(-2.5,-1)(0.4,0){13}
    {\line(1,0){0.2}}
  \put(0.2,1.4)
    {$\beta=v/c=\tanh\chi$}
  \qBezier(0,0)(0.8853,0.8853)
    (2,0.9640)
  \qBezier(0,0)(-0.8853,-0.8853)
    (-2,-0.9640)
  \put(-3,-2){\circle*{0.2}}
\end{picture}
```



Kontrolní body obou Bézierových křivek byly získány pomocí vzorců (5.1). Kladná část se určí pomocí $P_1 = (0, 0)$, $m_1 = 1$, $P_2 = (2, \tanh 2)$ a $m_2 = 1/\cosh^2 2$. Obrázek je opět definován pomocí vhodných souřadnic a levému dolnímu rohu jsou přiřazeny matematické souřadnice $(-3, -2)$, to je onen černý kotouč.

5.3. Grafický balík TikZ & PGF

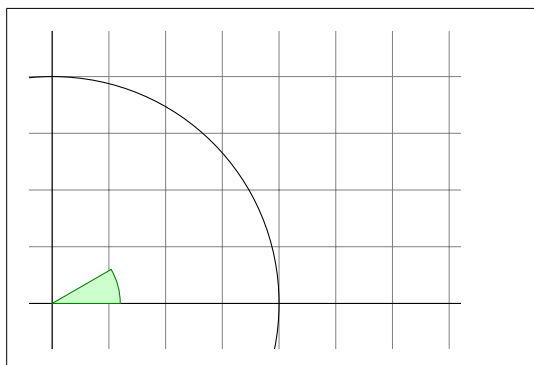
V dnešní době umí každý systém generující L^AT_EXový výstup vytvořit pěknou vektorovou grafiku, jen rozhraní se poněkud liší. Balík PGF poskytuje abstraktní vrstvu nad těmito rozhraními a umožňuje používat jednoduché příkazy pro vytvoření složité vektorové grafiky přímo „zevnitř“ L^AT_EXového dokumentu. Balík PGF obsahuje více než sedmisetstránkovou dokumentaci [18], my si však dovolíme být v této sekci o pár set stran stručnější.

Pro přístup k funkcím „vyšší úrovně“ balíku PGF byste si měli nahrát balík tikz, s kterým můžete používat výkonné příkazy ke kreslení grafiky přímo ze svého dokumentu. Kreslicí instrukce vložte dovnitř prostředí tikzpicture.

```

\begin{tikzpicture}[scale=3]
  \clip (-0.1,-0.2)
    rectangle (1.8,1.2);
  \draw
    [step=.25cm,gray,very thin]
    (-1.4,-1.4) grid (3.4,3.4);
  \draw (-1.5,0) -- (2.5,0);
  \draw (0,-1.5) -- (0,1.5);
  \draw (0,0) circle (1cm);
  \filldraw[fill=green!20!white,
    draw=green!50!black]
    (0,0) -- (3mm,0mm)
    arc (0:30:3mm) -- cycle;
\end{tikzpicture}

```



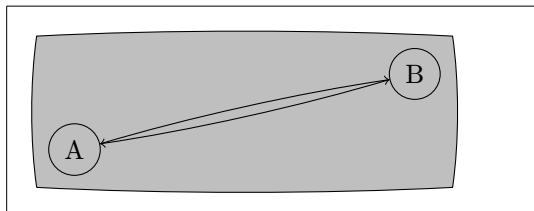
Pokud jste obeznáni s dalšími programovacími jazyky, možná si všimnete povědomého středníku (;) použitého k oddělení příkazů.

Pomocí příkazu `\usetikzlibrary` v preambuli můžete aktivovat řadu rozšiřujících rysů pro kreslení speciálních tvarů, např. tento box, který je trochu prohnutý.

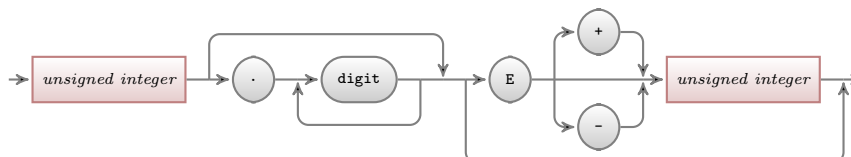
```

\usetikzlibrary{%
  decorations.pathmorphing}
\begin{tikzpicture}[
  decoration={bent,aspect=.3}]
  \draw [decorate,fill=lightgray]
    (0,0) rectangle (5.5,2);
  \node[circle,draw]
    (A) at (.5,.5) {A};
  \node[circle,draw]
    (B) at (5,1.5) {B};
  \draw[->,decorate] (A) -- (B);
  \draw[->,decorate] (B) -- (A);
\end{tikzpicture}

```



Můžete dokonce kreslit diagramy, které jako by vypadly z knihy o programování v Pascalu. Kód takového diagramu je rozsáhlejší než v předchozím příkladu, takže ukážeme jenom výsledek. V PGF dokumentaci je důkladný popis kresby tohoto diagramu.



Potřebujete-li kreslit grafy číselných hodnot nebo funkcí, měli byste se důkladněji seznámit s balíky `pgfplots` a `pgfplotstable`, které poskytují všechno potřebné. Vyhodnocení funkcí, které chcete zobrazit, umí dokonce udělat vyvoláním externího programu `gnuplot`, <http://www.gnuplot.info/>.

Pro ještě náročnější čtenáře zmiňujeme existenci jazyka R – výpočetní prostředí zaměřené na statistiku, <http://www.r-project.org/>, a pomocného balíku `Sweave`.

Nelze opominout též systém Sage – prostředí zaměřené na matematiku, <http://www.sagemath.org/>, a pomocný balík `SageTeX`, kde má našinec na vývoji a testování nemalé zásluhy (Robert Mařík, Brno).

Kapitola 6.

Přizpůsobování L^AT_EXu

Příkazy, které jsme už probrali, budou většině lidí stačit. I když vygenerované dokumenty nebudou elegantní, budou dodržovat zavedená pravidla sazby, takže se budou dobře číst a budou příjemné na pohled.

Stává se ale, že L^AT_EX neposkytuje příkaz nebo prostředí pro vaše potřeby, nebo vám nevyhovují výstupy vytvořené těmito prostředky.

V této kapitole naznačíme, jak L^AT_EX naučit novým trikům a jak docílit výstupů, které vypadají jinak než ty produkované standardně.

6.1. Nové příkazy, prostředí a balíky

Možná jste si všimli, že všechny příkazy v této knize jsou sázeny v boxu a že jsou uvedeny v indexu na konci knihy. Místo implementace pomocí základních L^AT_EXových příkazů jsem vytvořil balík, ve kterém jsem nadefinoval nové příkazy a prostředí. A tak teď můžu jednoduše napsat:

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



```
\dum
```

V tomto příkladě používám nové prostředí nazvané `lscommand`, které nakreslí okolo názvu příkazu box, a nový příkaz `\ci`, který vysází jméno příkazu neproporcionálním fontem se zpětným lomítkem a vytvoří příslušnou indexovou položku. Když se v indexu na konci knihy podíváte na položku `\dum`, najdete seznam stran, kde je příkaz `\dum` zmíněný.

Kdykoliv se rozhodnu, že jména příkazů v boxu sázet nechci, dosáhnu toho jednoduše změnou definice prostředí `lscommand`. To je mnohem jednodušší než být nucen projít celý dokument a hledat všechna místa, kde jsem použil obecný L^AT_EXový příkaz kreslící box okolo slova.

6.1.1. Nové příkazy

Své vlastní příkazy můžete přidávat pomocí příkazu

```
\newcommand{name}[num]{definition}
```

Tento příkaz vyžaduje dva argumenty: jméno (*name*) příkazu, který chcete vytvořit a jeho definici (*definition*). Argument *num* v hranatých závorkách je nepovinný a specifikuje počet argumentů nového příkazu (argumentů může být až devět). Pokud tento argument není zadán, bere se jeho hodnota jako 0, tj. nový příkaz nemá žádné argumenty.

Následující dva příklady ukazují použití. První z nich definuje příkaz `\tnss` (zkratka anglického názvu tohoto dokumentu: *The Not So Short Introduction to L^AT_EX 2_ε*). Podobný příkaz se vám bude hodit, budete-li potřebovat sázet název knihy na mnoha místech.

```
\newcommand{\tnss}{The not
  so Short Introduction to
  \LaTeXe}
This is ‘‘\tnss’’\ldots{}
```

```
This is “The not so Short Introduction
to LATEX 2ε”... “The not so Short Intro-
duction to LATEX 2ε”
```

Další příklad ukazuje, jak nadefinovat nový příkaz s jedním argumentem. Značka #1 bude nahrazena argumentem, který při použití příkazu specifikujete. Chcete-li použít více než jeden argument, můžete použít #2, #3, atd.

```
\newcommand{\txsit}[2]
  {Toto je \emph{#1} #2 úvod
  do \LaTeXe}
% v těle dokumentu:
\begin{itemize}
\item \txsit{nepříliš}{stručný}
\item \txsit{velmi}{obsáhlý}
\end{itemize}
```

- Toto je *nepříliš* stručný úvod do L^AT_EX 2_ε
- Toto je *velmi* obsáhlý úvod do L^AT_EX 2_ε

L^AT_EX vám nedovolí vytvořit nový příkaz, který by „přepsal“ příkaz už existující. Pokud ale existující příkaz opravdu přepsat chcete, můžete použít příkaz `\renewcommand`. Jeho syntax je stejná jako u příkazu `\newcommand`.

Může se vám hodit i další příkaz – `\providecommand`. Jeho použití je stejné jako u `\newcommand`, ale v případě, že příkaz už byl definován dříve, `\providecommand` nic neudělá.

Poznámky o mezerách následujících za L^AT_EXovými příkazy si můžete prohlédnout na straně 5.

6.1.2. Nová prostředí

Stejně jako můžete vytvářet nové příkazy (pomocí `\newcommand`), můžete vytvářet i svá vlastní prostředí. Slouží k tomu příkaz `\newenvironment`, jehož syntax je:

```
\newenvironment{name}[num]{before}{after}
```

I `\newenvironment` může mít nepovinný parametr. Materiál specifikovaný v argumentu *before* (před) bude zpracován před tím, než se zpracuje text v prostředí. Materiál v argumentu *after* (za) bude zpracován ve chvíli, kdy \LaTeX narazí na příkaz `\end{name}`.

Následující příklad ukazuje použití příkazu `\newenvironment`.

```
\newenvironment{kral}
  {\rule{1ex}{1ex}%
   \hspace{\stretch{1}}}
  {\hspace{\stretch{1}}%
   \rule{1ex}{1ex}}
```

■ Mí ubozí poddaní... ■

```
\begin{kral}
Mí ubozí poddaní\ldots
\end{kral}
```

Argument *num* má stejný význam jako u příkazu `\newcommand`. \LaTeX vás nenechá nedefinovat prostředí, které už existuje. Pokud opravdu chcete změnit existující prostředí, použijte příkaz `\renewenvironment`. Jeho syntax je stejná jako u příkazu `\newenvironment`.

Příkazy použité v tomto příkladu budou vysvětleny později, konkrétně příkaz `\rule` na straně 131, `\stretch` na straně 125 a `\hspace` na straně 124.

6.1.3. Mezery navíc

Při definování nového prostředí se vám snadno může stát, že do dokumentu „zavlečete“ mezery, které tam nepatří. Tyto mezery můžou způsobit velké komplikace, např. pokud chcete vytvořit prostředí pro nadpisy, které potlačí odsazení jak v samotném nadpisu, tak v (jednom) následujícím odstavci. Příkaz `\ignorespaces` v bloku `\begin` daného prostředí způsobí, že budou ignorovány všechny mezery, které se vyskytnou po provedení bloku `\begin`. U bloku `\end` je situace trochu komplikovanější, protože na konci prostředí dochází ke speciálnímu zpracování. Po použití příkazu `\ignorespacesafterend` \LaTeX sám vyvolá `\ignorespaces` po skončení zmíněného speciálního ‚závěrečného‘ zpracování.

```
\newenvironment{jednoduche}%
  {\noindent}%
  {\par\noindent}

\begin{jednoduche}
Všimněte si mezery\\vlevo.
\end{jednoduche}
To samé\\zde.
```

<p>Všimněte si mezery vlevo.</p> <p>To samé zde.</p>
--

```
\newenvironment{spravne}%
  {\noindent\ignorespaces}%
  {\par\noindent%
   \ignorespacesafterend}

\begin{spravne}
Žádná mezera\\vlevo.
\end{spravne}
To samé\\zde.
```

<p>Žádná mezera vlevo.</p> <p>To samé zde.</p>
--

6.1.4. Příkazová řádka L^AT_EXu

Pokud L^AT_EX používáte na UNIXovém operačním systému, možná si práci automatizujete prostřednictvím makefilů. V tom případě pro vás může být zajímavé automatizovaně produkovat různé verze stejného dokumentu pomocí volání L^AT_EXu s parametry předávanými na příkazové řádce. Pokud do svého dokumentu přidáte následující strukturu...

```
\usepackage{ifthen}
\ifthenelse{\equal{\blackandwhite}{true}}{
  % "černobílý" režim; zde něco napíšme..
}{
  % "barevný" režim; zde napíšme něco jiného..
}
```

... a potom L^AT_EX spustíte takto...

```
latex '\newcommand{\blackandwhite}{true}\input{test.tex}'
```

... pak L^AT_EX nejdříve nadefinuje příkaz `\blackandwhite` a potom načte vlastní soubor (`test.tex`) příkazem `\input`. Nastavením `\blackandwhite` na hodnotu `false` zařídíme, že se vytvoří barevná verze dokumentu.

6.1.5. Váš vlastní balík

Pokud si nadefinujete hodně nových prostředí a příkazů, preambule vašeho dokumentu se pořádně prodlouží. Je proto lepší vytvořit L^AT_EXový balík,

```
% Demo Package by Tobias Oetiker
\ProvidesPackage{demopack}
\newcommand{\tnss}{The not so Short Introduction
to \LaTeXe}
\newcommand{\txsit}[1]{The \emph{#1} Short
Introduction to \LaTeXe}
\newenvironment{king}{\begin{quote}}{\end{quote}}
```

Obrázek 6.1: Pokusný balík.

který bude obsahovat definice všech vašich příkazů a prostředí. Pomocí `\usepackage` potom můžete tento balík ve svém dokumentu zpřístupnit.

Vytvoření balíku v zásadě znamená zkopírování obsahu preambule vašeho dokumentu do zvláštního souboru, jehož jméno končí na `.sty`. Speciální příkaz...

`\ProvidesPackage{package name}`

použijete na úplném začátku souboru vašeho balíku. `\ProvidesPackage` říká \LaTeX u jméno balíku a \LaTeX tak bude moci zobrazit smysluplnou chybovou hlášku v případě, že se pokusíte balík vložit do dokumentu dvakrát. Na obrázku 6.1 je malý pokusný balík, který obsahuje příkazy definované v předchozích příkladech.

6.2. Fonty a velikosti

6.2.1. Příkazy na změnu fontů

\LaTeX vybere příslušný font a jeho velikost na základě logické struktury dokumentu (sekce, poznámky pod čarou, ...). Může se ale stát, že budete chtít změnit font nebo jeho velikost „manuálně“. Můžete toho dosáhnout pomocí příkazů uvedených v tabulkách 6.1 a 6.2. Skutečná velikost každého fontu je otázkou designu a závisí na třídě dokumentu a jejích nastaveních. Tabulka 6.3 ukazuje absolutní (bodové) velikosti pro tyto příkazy tak, jak jsou implementovány ve standardních třídách dokumentu.

```
{\small Malí
a~\textbf{smělí} Římané vládli}
{\Large celé velké
\textit{Itálii}.}
```

Malí a **smělí** Římané vládli celé velké
Itálii.

Důležitým rysem L^AT_EX 2_ε je, že atributy fontů jsou nezávislé. Můžete změnit aktuální font nebo jeho velikost, ale nastavení tučnosti nebo sklonu písma zůstane nezměněno.

V *matematickém módu* můžete pomocí *příkazů* měnících font dočasně opustit *matematický režim* a zapsat obyčejný text. Pokud chcete změnit font pro zápis matematiky, budete potřebovat jinou speciální sadu příkazů, viz tabulka 6.4.

Nejen v souvislosti s příkazy měnícími velikost fontů hrají zvláštní roli složené závorky, pomocí nichž se vytváří *skupiny*. Skupiny omezují rozsah platnosti většiny L^AT_EXových příkazů.

Má rád `{\LARGE}` velká
a `{\small}` malá písмена.

Má rád **velká** a malá písмена.

Příkazy pro změnu velikosti fontu také mění mezery mezi řádky odstavce, ale jenom tehdy, když daný odstavec skončí dříve než rozsah platnosti příkazu pro změnu velikosti fontu. Závírací složená závorka `}` by tedy neměla

Tabulka 6.1: Fonty.

<code>\textrm{...}</code>	vzpřímené	<code>\textsf{...}</code>	bezpatkové
<code>\texttt{...}</code>	psací stroj		
<code>\textmd{...}</code>	střední	<code>\textbf{...}</code>	tučné
<code>\textup{...}</code>	vzpřímené písmo	<code>\textit{...}</code>	<i>italika</i>
<code>\textsl{...}</code>	<i>skloněné písmo</i>	<code>\textsc{...}</code>	KAPITÁLKY
<code>\emph{...}</code>	<i>zvýrazněné</i>	<code>\textnormal{...}</code>	základní font

Tabulka 6.2: Velikosti fontů.

<code>\tiny</code>	drobný font	<code>\Large</code>	větší font
<code>\scriptsize</code>	velmi malý font	<code>\LARGE</code>	velmi velký
<code>\footnotesize</code>	docela malý font		
<code>\small</code>	malý font	<code>\huge</code>	obří
<code>\normalsize</code>	normální font	<code>\Huge</code>	největší
<code>\large</code>	velký font		

Tabulka 6.3: Absolutní bodové velikosti ve standardních třídách.

Velikost	10pt (výchozí)	volba 11pt	volba 12pt
<code>\tiny</code>	5pt	6pt	6pt
<code>\scriptsize</code>	7pt	8pt	8pt
<code>\footnotesize</code>	8pt	9pt	10pt
<code>\small</code>	9pt	10pt	11pt
<code>\normalsize</code>	10pt	11pt	12pt
<code>\large</code>	12pt	12pt	14pt
<code>\Large</code>	14pt	14pt	17pt
<code>\LARGE</code>	17pt	17pt	20pt
<code>\huge</code>	20pt	20pt	25pt
<code>\Huge</code>	25pt	25pt	25pt

Tabulka 6.4: Matematické fonty.

<code>\mathrm{...}</code>	Zkladn patkov font
<code>\mathbf{...}</code>	Tun font
<code>\mathsf{...}</code>	Bezpatkov font
<code>\mathtt{...}</code>	Font psacho stroje
<code>\mathit{...}</code>	<i>Italika</i>
<code>\mathcal{...}</code>	<i>KALIGRAFIJK FONT</i>
<code>\mathnormal{...}</code>	<i>Normln font</i>

být zapsána „příliš brzy“. V následujících příkladech si všimněte umístění příkazu `\par`.¹

```
{\Large Toto nečtěte!  
Není to pravda.  
Můžete mi věřit!}\par}
```

Toto nečtěte! Není to pravda.
Můžete mi věřit!

```
{\Large Ani toto není pravda.  
Ale nezapomeňte,  
že jsem lhář.}\par}
```

Ani toto není pravda. Ale nezapomeňte, že jsem lhář.

Pokud chcete aktivovat příkaz pro změnu velikosti fontu pro celý odstavec nebo ještě větší úsek, může se vám hodit syntaxe pro prostředí.

```
\begin{Large}  
Toto není pravda.  
Na druhou stranu,  
co dnes je\ldots  
\end{Large}
```

Toto není pravda. Na druhou stranu, co dnes je...

Ušetříte si počítání spousty složených závorek.

6.2.2. Pozor, pozor

Jak jsme řekli na začátku této kapitoly, není dobré zahltit dokument explicitními příkazy typu změny velikosti fontu. To je totiž v protikladu k základnímu přístupu L^AT_EXu, který odděluje logické a vizuální značení dokumentu. Správné naopak je, chcete-li použít stejný příkaz pro změnu fontu na více místech v dokumentu k vysázení speciálního druhu informace, nadefinovat pomocí `\newcommand` „příkaz – logický obalovač“ pro příkaz změny fontu.

```
\newcommand{\oops}[1]{%  
  \textbf{#1}}  
\oops{Nevstupujte do tohoto  
cizího pokoje, jsou v něm  
\oops{stroje} neznámého  
původu a účelu.}
```

Nevstupujte do tohoto cizího pokoje, jsou v něm **stroje** neznámého původu a účelu.

Tento přístup má tu výhodu, že když se někdy později rozhodnete, že budete nebezpečí značit jinak než pomocí `\textbf`, stačí v dokumentu udělat jedinou změnu – v definici příkazu `\oops`. Není tedy nutné projít všechny výskyty `\textbf` v celém dokumentu a pro každý z nich zkoumat, jestli bylo důvodem zvýraznění daného textu nebezpečí, nebo něco jiného.

¹Příkaz `\par` má stejný účinek jako prázdný řádek.

6.2.3. Rada

Malá rada na závěr tohoto výletu do země fontů a jejich velikostí:

Nezapomeňte! Čím **ví** *CE* fontů v dokumentu
použijete, *tím* ČITELNĚJŠÍ a *příjemnější na pohled* bude. :-)

6.3. Mezery

6.3.1. Řádkové mezery

Chcete-li mít ve svém dokumentu větší řádkové mezery, uveďte v preambuli příkaz

```
\linespread{factor}
```

Použijte `\linespread{1.3}` nebo `\linespread{1.6}` pro řádkování „jeden a půl“, resp. pro dvojnásobné řádkování. Implicitně řádkové mezery roztažené nejsou (jejich „faktor roztažení“ je 1).

Všimněte si, že příkaz `\linespread` je docela drastický a pro publikování se moc nehodí. Pokud opravdu máte důvod změnit velikost řádkových mezer, mohli byste spíš použít příkaz:

```
\setlength{\baselineskip}{1.5\baselineskip}
```

```
{\setlength{\baselineskip}%  
{1.5\baselineskip}}
```

Tento odstavec je vysázen s meziřádkovými mezerami roztaženými jedenapůlkrát oproti předchozímu stavu. Všimněte si příkazu `par` na konci odstavce. `\par}`

Účelem tohoto odstavce je ukázat, že po uzavírací složené závorce se všechno vrací zpět do normálu.

Tento odstavec je vysázen s meziřádkovými mezerami roztaženými jedenapůlkrát oproti předchozímu stavu. Všimněte si příkazu `par` na konci odstavce.

Účelem tohoto odstavce je ukázat, že po uzavírací složené závorce se všechno vrací zpět do normálu.

6.3.2. Formátování odstavce

Dva \LaTeX ové parametry ovlivňují vzhled odstavce. Napsáním definice podobné této

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

do preambule vašeho vstupního souboru můžete změnit vzhled odstavců. Tyto dva příkazy zvětší mezery mezi dvěma odstavci a zároveň nastaví velikost odsazení prvního řádku odstavce na nulu.

plus a minus ve specifikaci délky výše říkájí T_EXu, že délka meziodstavcových mezer může být stažena nebo roztažena o specifikované množství. Ke stažení nebo roztažení mezer dojde ve chvíli, kdy by odstavec s neroztaženými mezerami nešlo správně vysázet.

V kontinentální Evropě je zvykem odstavce oddělovat vertikálními mezerami a neodsazovat první řádek odstavce. Ale pozor, změnou vertikálních mezer ovlivníte i obsah, jehož řádky budou pak mít větší mezery. Lze se tomu vyhnout tím, že přesuneme dva zmíněné příkazy z preambule někam do vašeho dokumentu za příkazem `\tableofcontents`. Možná ještě lépe zvětšení mezer v obsahu zabráníte tím, že výše zmíněné dva příkazy vůbec nepoužijete. Většina profesionálních publikací totiž mezery mezi odstavce nekládá a místo toho odsazuje první řádek.

Chcete-li odsadit první řádek odstavce, který by jinak odsazen nebyl, začněte odstavec

```
\indent
```

Tato odsazení budou samozřejmě mít efekt pouze tehdy, má-li `\parindent` nenulovou hodnotu.²

Neodsazované odstavce vytvoříte uvedením

```
\noindent
```

na úplném začátku odstavce. To se může hodit např. tehdy, když zahajujete tělo dokumentu „obyčejným textem“ a ne novou sekci.

6.3.3. Horizontální mezery

L^AT_EX stanovuje mezery mezi slovy a větami automaticky. Přidat horizontální mezeru manuálně lze pomocí

```
\hspace{length}
```

Pokud se má daná mezerka vysázet i tehdy, když případně na konec nebo začátek řádku, použijte `\hspace*` místo `\hspace`. Parametr *length* je v nejjednodušším případě jen číslo následované jednotkou. Nejdůležitější jednotky jsou uvedené v tabulce 6.5.

²K odsazení prvního odstavce sekce, použijte balík `indentfirst` z kolekce `,tools'`.

Tabulka 6.5: T_EXové jednotky.

mm	milimetr $\approx 1/25$ inch	⊐
cm	centimetr = 10 mm	┌───┐
in	palec = 25.4 mm	┌──────────┐
pt	bod $\approx 1/72$ inch $\approx \frac{1}{3}$ mm	⊐
em	zhruba šířka písmene ‘M’ v aktuálním fontu	┌┐
ex	zhruba výška písmene ‘x’ v aktuálním fontu	┐┐

Toto `\hspace{1.5cm}` je mezera velikosti 1.5 cm.

Toto je mezera velikosti 1.5 cm.

Příkaz

`\stretch{n}`

vytváří speciální „natahovací“ mezery. Mezera se natáhne až do konce aktuálního řádku. Pokud na stejném řádku uvedeme `\hspace{\stretch{n}}` několikrát, natahovací mezery si rozdělí všechny zbývající prostor na řádku v poměru jejich natahovacích faktorů.

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

x x x

Když používáme horizontální mezery společně s textem, může být výhodné velikost mezer specifikovat v závislosti na velikosti aktuálního fontu. Toho lze dosáhnout použitím „relativních jednotek“ `em` a `ex`.

`{\Large}velká\hspace{1em}y\`
`{\tiny}malá\hspace{1em}y`

velká y
malá y

6.3.4. Vertikální mezery

I mezery mezi odstavci, sekcemi, podsekcemi, ... určí L^AT_EX automaticky. Dodatečné vertikální mezery *mezi dvěma odstavci* lze přidat příkazem:

`\vspace{length}`

Použijete-li tento příkaz, měli byste ho uvést mezi dvěma prázdnými řádky. Pokud chcete, aby tato mezera nezmizela v případě, že se ocitne na

samém začátku nebo konci stránky, použijte ohvězdičkovanou verzi příkazu, `\vspace*`, místo `\vspace`.

Příkaz `\stretch`, v kombinaci s příkazem `\pagebreak`, lze použít pro umístění textu na poslední řádku na stránce nebo pro vertikální vycentrování textu na stránce.

Nějaký text\ldots

Příkaz `\verb|\stretch|`, v kombinaci s příkazem `\verb|\pagebreak|`, lze použít pro umístění textu na poslední řádku na stránce nebo pro vertikální vycentrování textu na stránce. Nějaký text\ldots

`\vspace{\stretch{1}}`

Toto bude vysázeno na poslední řádku na stránce\pagebreak

Dodatečnou mezeru mezi dvěma řádky *stejného* odstavce nebo uvnitř tabulky získáme pomocí příkazu

`\\[length]`

Pomocí příkazů `\bigskip`, `\medskip` a `\smallskip` můžete přeskočit oblast o předdefinované vertikální velikosti, aniž byste se museli starat o přesné rozměry.

6.4. Layout stránky

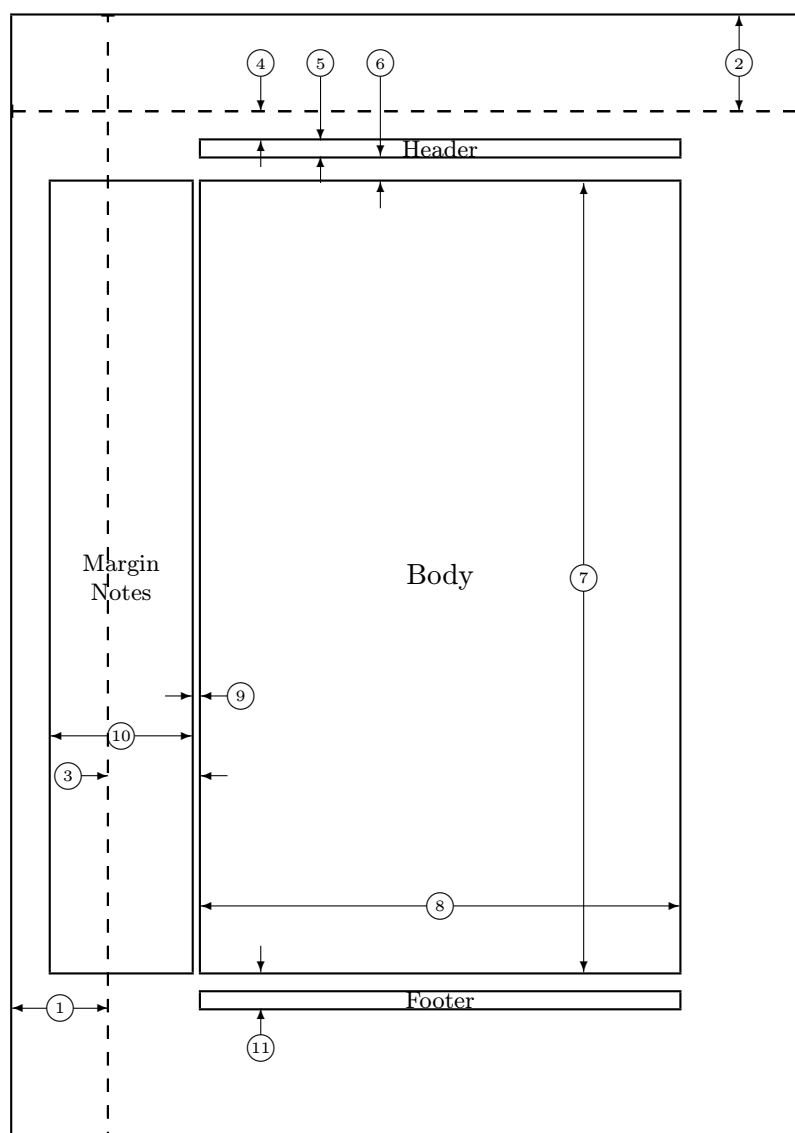
L^AT_EX 2_ε vám umožňuje v příkazu `\documentclass` specifikovat velikost papíru. L^AT_EX 2_ε sám pak automaticky použije správné velikosti okrajů stránky. Pokud vám předdefinované hodnoty nevyhovují, můžete je samozřejmě změnit. Obrázek 6.2 ukazuje všechny parametry, které můžete změnit. Obrázek byl vytvořen pomocí balíku `layout` z kolekce `tools`.³

Ale pozor! Předtím, že začnete svůj dokument „vylepšovat“ ve stylu „Jen trochu rozšířím tu úzkou stránku“, všechno si promyslete. Podobně jako další standardní L^AT_EXová nastavení má i standardní vzhled stránky svůj dobrý důvod.

Ano, v porovnání se stránkou textu napsanou v Microsoft Wordu vypadá vaše L^AT_EXová stránka příliš úzká. Ale zato se nápadně podobá standardně sázeným knihám.⁴ Podobně jako ony má i váš dokument zhruba šedesát šest znaků na řádku, což je hodnota standardně považovaná za hranici, po jejímž překročení se dokument špatně čte. Důvodem je, že je obtížné přesunout oči

³`macros/latex/required/tools`

⁴Máme na mysli knihu od solidního nakladatele.



1	one inch + <code>\hoffset</code>	2	one inch + <code>\voffset</code>
3	<code>\oddsidemargin = 22pt</code> or <code>\evensidemargin</code>	4	<code>\topmargin = 22pt</code>
5	<code>\headheight = 12pt</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> <code>\marginparpush = 5pt</code> (not shown)
11	<code>\footskip = 27pt</code> <code>\hoffset = 0pt</code> <code>\paperwidth = 597pt</code>		<code>\voffset = 0pt</code> <code>\paperheight = 845pt</code>

Obrázek 6.2: Parametry rozložení stránky.

z konce jednoho řádku na začátek toho následujícího. Proto také bývá text v novinách dělený do sloupců.

Pokud tedy zvětšíte šířku vašich řádek, nezapomínejte, že tím můžete komplikovat čtení zájemcům o váš dokument. Ale teď už dost varování, slíbili jsme, že vám řekneme, jak text rozšířit. . .

L^AT_EX poskytuje dva příkazy pro změnu těchto parametrů. Obvykle se tyto příkazy uvádějí v preambuli dokumentu.

První příkaz do libovolného z parametrů přiřadí pevnou hodnotu:

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

Druhý příkaz zvětší hodnotu libovolného parametru:

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

Tento druhý příkaz je užitečnější než příkaz `\setlength`, protože jeho efekt je relativní vzhledem k existujícím hodnotám. Pro rozšíření celkové šířky textu jsem uvedl následující příkaz do preambule dokumentu:

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

V tomto kontextu by se vám mohl hodit balík `calc`. Ten umožňuje použít aritmetické výrazy jak v argumentu `\setlength` tak v ostatních místech, kde se v argumentech předávají číselné hodnoty.

6.5. Další hrátky s délkami

Absolutním délkám se v L^AT_EXovém dokumentu snažím vyhnout. Délky raději odvozují ze šířky nebo výšky dalších stránkových elementů. Např. šířka obrázku specifikovaná jako `1\textwidth` znamená „jedenkrát přes celou šířku stránky“.

Následující tři příkazy nastavují šířku, výšku a hloubku textového řetězce *text* do délkového rozměru *variable*.

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

Následující příklad ukazuje jedno z možných použití těchto příkazů.

```

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

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

\begin{vardesc}{Kde}$a$,
$b$ -- jsou strany sousedící
s pravým úhlem pravouhlého
trojúhelníku.

$c$ -- je přepona tohoto
trojúhelníku a cítí se osamělá.

$d$ -- se zde nakonec vůbec
neobjevuje. To je divné, ne?
\end{vardesc}

```

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

Kde: a , b – jsou strany sousedící
s pravým úhlem pravouhlého
trojúhelníku.

c – je přepona tohoto trojúhelníku
a cítí se osamělá.

d – se zde nakonec vůbec
neobjevuje. To je divné, ne?

Odsazení prvních řádků se nastaví na délku slova „Kde:␣“, tedy c a d jsou vysázeny pod písmenkem a v části „ a , b “.

6.6. Boxy

\LaTeX vytváří stránky dokumentu posunováním boxů. Každé písmeno je nejdříve malým boxem. Boxy s písmeny se potom spojují k sobě a vznikají slova. Slova jsou potom spojována s dalšími slovy, a to pomocí speciálních, „elastických“ mezer. Tyto elastické mezery \LaTeX může natahovat nebo zkracovat, pokud mu to umožní vysázet daný text lépe.

Toto je velmi zjednodušený popis toho, co se při sazbě opravdu děje, ale podstatné je, že \TeX pracuje s elastickými mezerami a boxy. Písmena nejsou jediné věci, které se mohou ocitnout v boxu. Do boxu můžete umístit téměř všechno, včetně dalších boxů. S každým boxem potom \LaTeX zachází jako s jednotlivým písmenem.

V předchozích kapitolách jste se už s boxy setkali, jen jsem vás na ně neupozornil. Boxy jsou produkovány např. prostředím `tabular` nebo příkazem `\includegraphics`. Díky tomu můžete jednoduše umístit dvě tabulky nebo obrázky vedle sebe. Musíte jen zajistit, že dohromady jejich šířky nepřekročí `\textwidth`.

Pomocí příkazu

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

nebo prostředí

```
\begin{minipage}[pos]{width} text \end{minipage}
```

můžete do boxu umístit odstavec textu. Parametr `pos` určuje vertikální zarovnání boxu vzhledem k účarí obklopujícího textu a může nabývat jedné ze tří hodnot: `c`, `t` nebo `b`.⁵ `width` je délka určující šířku boxu. Hlavní rozdíl mezi `minipage` a `\parbox` je ten, že uvnitř `parbox` nemůžete použít některé příkazy a prostředí, zatímco uvnitř `minipage` můžete použít skoro vše.

Zatímco `\parbox` zabalí do boxu celý odstavec, který předtím rozdělí na jednotlivé řádky textu, existuje také skupina příkazů pro umístění materiálu do boxu, která pracuje jen s horizontálně zarovnaným materiálem. Jeden z těchto příkazů už známe – `\mbox`. Tento příkaz jednoduše zabalí posloupnost boxů do nového boxu. `\mbox` můžeme použít, abychom L^AT_EXu zabránili rozdělit od sebe dvě slova. Díky tomu, že boxy mohou být umístěny do dalších boxů, tyto příkazy pro „zaboxování“ horizontálního materiálu jsou velmi flexibilní.

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

`width` definuje šířku výsledného boxu tak, jak bude „vidět zvenku“.⁶ Jako parametr `width` můžete kromě dimenzních výrazů předat i `\width`, `\height`, `\depth` nebo `\totalheight`, jejichž hodnota je nastavena změřením vysázeného `text`. V parametru `pos` se předává jednopísmenná hodnota: `center`, `flushleft` (zarovnej doleva), `flushright` (zarovnej doprava), nebo `spread` (zarovnej text tak, aby vyplnil celý box).

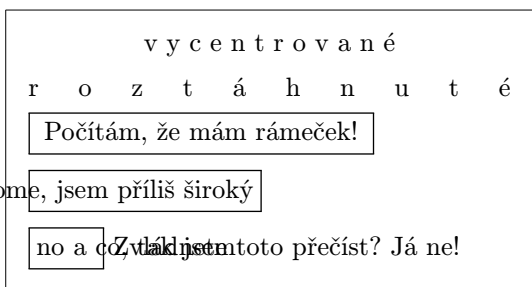
Příkaz `\framebox` dělá to samé jako `\makebox`, ale navíc ještě okolo textu nakreslí rámeček.

Některé způsoby, jak příkazy `\makebox` a `\framebox` použít, ukazuje následující příklad.

⁵Parametry jsou: `center` (střed), `top` (horní strana), `bottom` (spodní strana).

⁶Tato definovaná šířka může být menší než šířka materiálu vytvořeného z `text`. Můžete dokonce nastavit `width` na `0pt` a docílit tím toho, že vysázení textu uvnitř boxu neovlivní okolní boxy, tedy posun referenčního bodu.


```
\makebox[\textwidth]{%
  v~y c~e~n t~r~o v~a n~e~}\par
\makebox[\textwidth][s]{%
  r~o~z t~a~h n~u t~e~}\par
\framebox[1.1\width]{Počítám, že
  mám rámeček!} \par
\framebox[0.8\width][r]{Hrome, Hrome,
  jsem příliš široký} \par
\framebox[1cm][l]{no a~co,
  tak jsem}
Zvládnete toto přečíst? Já ne!
```

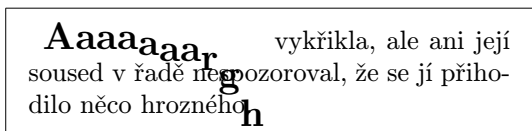


Horizontální efekty jsme se tedy právě naučili, teď se podívejme na vertikální.⁷ Příkaz

```
\raisebox{lift}[extend-above-baseline][extend-below-baseline]{text}
```

vám umožní nadefinovat vnitřní vlastnosti boxu. Pokud se v prvních třech parametrech potřebujete odkázat na šířku, výšku nebo hloubku aktuálního boxu, můžete použít `\width`, `\height`, `\depth` nebo `\totalheight`.

```
\raisebox{0pt}[0pt][0pt]{\Large%
\textbf{
Aaaa\raisebox{-0.3ex}{a}%
\raisebox{-0.7ex}{aa}%
\raisebox{-1.2ex}{r}%
\raisebox{-2.2ex}{g}%
\raisebox{-4.5ex}{h}}}
vykřikla, ale ani její souseď
v řadě nespozoroval, že se jí
přihodilo něco hrozného.
```



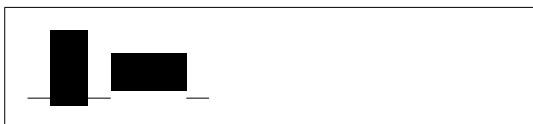
6.7. Linky

O několik stránek dříve jste si možná všimli příkazu

```
\rule[lift]{width}{height}
```

Jeho pomocí se obvykle kreslí jednoduchý černý box.

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



⁷Pro úplnou kontrolu nad boxy potřebujeme jak horizontální, tak vertikální efekty.

To se hodí pro kreslení vertikálních nebo horizontálních čar. Čáru na titulní stránce jsme např. nakreslili právě pomocí tohoto příkazu.

Konec knihy.
Hodně zdaru při \TeX ování!

Příloha A.

Instalace L^AT_EXu

Donald E. Knuth publikoval zdrojové kódy T_EXu v době, kdy ještě nikdo neslyšel o OpenSource a/nebo Free Software. Licence, kterou je T_EX opatřen, dovoluje, abyste zdrojové kódy T_EXu použili jakýmkoliv způsobem. Pokud ale zdrojové kódy upravíte a vytvoříte z nich nový systém, můžete ho nazývat T_EX jen v případě, že výsledný program uspěje v sadě testů, které Knuth dal také k dispozici. Díky tomu máme dnes T_EXové implementace zadarmo pro téměř každý známý operační systém.

V této příloze naznačíme, co je pro zprovoznění T_EXu na Linuxu, Mac OS X nebo Microsoft Windows potřeba nainstalovat.

A.1. Co instalovat

Ať chcete L^AT_EX používat na jakémkoliv počítači, budete potřebovat čtyři programy:

1. Textový editor, v kterém budete vytvářet a upravovat vaše L^AT_EXové zdrojové soubory.
2. Program T_EX/L^AT_EX, který vaše L^AT_EXové zdrojové soubory vysází, tj. převede do formátu PDF nebo DVI.
3. Prohlížeč PDF/DVI – program, který umožňuje prohlížet a tisknout vaše dokumenty.
4. Program, který umožní přidávat do vašeho dokumentu Postscriptové soubory a obrázky.

Pro každou platformu je k dispozici mnoho takových programů. Zde zmíníme jen ty, které jsme si vyzkoušeli a líbí se nám.

A.2. T_EX na Mac OS X

A.2.1. Volba editoru

Založte své L^AT_EXové prostředí na editoru TextMate! TextMate je vysoce konfigurovatelný textový editor pro obecné použití. Navíc skvělým způsobem podporuje L^AT_EX a je úzce zintegrován s prohlížečím programem PDFView. Tato kombinace vám umožní používat L^AT_EX šikovným „Macovým“ způsobem. Zkušební verzi si můžete zadarmo stáhnout z domovské stránky TextMate na adrese <http://macromates.com/>, plná verze stojí 39 EUR. Znáte-li podobný OpenSource nástroj pro Mac, dejte nám prosím vědět.

A.2.2. Obstarejte si T_EXovou distribuci

Pokud pro instalování programů na OS X používáte Macports nebo Fink, použijte tyto manažery balíků i na instalaci L^AT_EXu. Uživatelé Macportu nainstalují L^AT_EX pomocí `port install tetex`, uživatelé Finku použijí příkaz `fink install tetex`.

Pokud Macports ani Fink nepoužíváte, stáhněte si MacT_EX, což je zkompileovaná L^AT_EXová distribuce pro OS X. MacT_EX poskytuje plnou L^AT_EXovou instalaci a navíc řadu přídatných nástrojů. MacT_EX můžete získat na adrese <http://www.tug.org/mactex/>.

A.2.3. Dopřejte si PDFView

Pro prohlížení PDF souborů vygenerovaných L^AT_EXem použijte PDFView. Tento program je úzce integrován s vaším L^AT_EXovým textovým editorem. PDFView je open-source aplikace a můžete si ho stáhnout z domovské stránky <http://pdfview.sourceforge.net/>. Po nainstalování se ujistěte, že v dialogu *preferences* programu je aktivováno nastavení *automatically reload documents* a že je nastavená podpora pro TextMate.

A.3. T_EX na Windows

A.3.1. Obstarání T_EXu

Nejdříve ze všeho si z adresy <http://www.miktex.org/> obstarejte distribuci MiK_TE_X. Obsahuje všechny základní programy a soubory potřebné pro přeložení L^AT_EXových dokumentů. Já osobně jsem unešen z toho, že MiK_TE_X je schopen automaticky stáhnout a nainstalovat chybějící L^AT_EXové balíky během zpracování dokumentu.

A.3.2. L^AT_EXový editor

L^AT_EX je programovacím jazykem pro textové dokumenty. T_EXnicCenter používá mnoho konceptů z programovacího světa a díky tomu poskytuje elegantní a efektivní prostředí pro editování L^AT_EXových zdrojových souborů na Windows. Můžete si ho obstarat na <http://www.toolscenter.org/>. T_EXnicCenter je šikovně zintegrováno s MiK_TE_Xem.

Další výbornou volbou je editor projektu L^ED, dostupný na serveru <http://www.latexeditor.org/>.

A.3.3. Práce s grafikou

Chcete-li v dokumentu zpracovaném L^AT_EXem použít grafiku vysoké kvality, obrázky, které do dokumentu přidáváte musí být ve formátu Postscript (eps) nebo PDF. Program GhostScript lze použít pro práci s těmito obrázkovými formáty a můžete ho získat, spolu s náhledovým programem GhostView, z adresy <http://www.cs.wisc.edu/~ghost/>.

Pro bitmapovou grafiku (fotky nebo naskenované obrázky) můžete použít program Gimp – open source alternativu k programu Photoshop. Program získáte na adrese <http://gimp-win.sourceforge.net/>.

Pro vektorovou grafiku zkuste Inkscape, <http://inkscape.org/>.

A.4. T_EX na Linuxu

L^AT_EX bývá často instalován spolu s operačním systémem Linux, nebo je minimálně dostupný ve zdroji vaší instalace. Pomocí svého manažera balíků si nainstalujte následující:

- tetex nebo texlive (texlive-full) – základní T_EXové/L^AT_EXové nastavení.
- emacs (s auctex) – Linuxový editor, který je s L^AT_EXem úzce integrován pomocí přídatného balíku AucT_EX.
- ghostscript – program na prohlížení a tisknutí PostScriptových dokumentů.
- xpdf a acrobat – program na prohlížení a tisknutí PDF dokumentů.
- imagemagick – volně dostupný program pro konvertování bitmapových obrázků.
- gimp – volně dostupná alternativa k programu Photoshop.
- inkscape – volně dostupná alternativa k programům Illustrator a Corel.

A.5. Projekt \TeX onWeb

Pokud nemůžete na počítači instalovat (např. nemáte administrátorská práva) ani si spustit \TeX z DVD Live, vyzkoušejte projekt \TeX onWeb (autor Jan Přichystal, Brno), <http://tex.mendelu.cz/>.

Literatura

- [1] Leslie Lamport. *L^AT_EX: A Document Preparation System*. Addison-Wesley, Reading, Massachusetts, druhé vydání, 1994. ISBN 0-201-52983-1.
- [2] Donald E. Knuth. *The T_EXbook*, Volume A of *Computers and Typesetting*, Addison-Wesley, Reading, Massachusetts, druhé vydání, 1984, ISBN 0-201-13448-9.
- [3] Frank Mittelbach, Michel Goossens, Johannes Braams, David Carlisle, Chris Rowley. *The L^AT_EX Companion*, druhé vydání. Addison-Wesley, Reading, Massachusetts, 2004. ISBN 0-201-36299-6.
- [4] Michel Goossens, Sebastian Rahtz and Frank Mittelbach. *The L^AT_EX Graphics Companion*. Addison-Wesley, Reading, Massachusetts, 1997. ISBN 0-201-85469-4.
- [5] Součástí každé L^AT_EXové instalace by měl být tzv. *lokální L^AT_EXový průvodce* – příručka vysvětlující zvláštnosti příslušné verze systému. Průvodce by měl být uložen v souboru `local.tex`. Pokud tento dokument ve vaší instalaci chybí, nezbude vám, než se s žádostí o pomoc obrátit na místního L^AT_EXového guru. Případně si otevřete příručku k [T_EX Live](#) nebo [L^AT_EX manual](#).
- [6] L^AT_EX3 Project Team. *L^AT_EX 2_ε for authors*. Je součástí distribuce L^AT_EX 2_ε – soubor `usrguide.tex`.
[CTAN://tex-archive/macros/latex/doc/usrguide.pdf](http://tex-archive/macros/latex/doc/usrguide.pdf)
- [7] L^AT_EX3 Project Team. *L^AT_EX 2_ε for Class and Package writers*. Je součástí distribuce L^AT_EX 2_ε – soubor `clsguide.tex`.
[CTAN://tex-archive/macros/latex/doc/clsguide.pdf](http://tex-archive/macros/latex/doc/clsguide.pdf)
- [8] L^AT_EX3 Project Team. *L^AT_EX 2_ε Font selection*. Je součástí distribuce L^AT_EX 2_ε – soubor `fntguide.tex`.
[CTAN://tex-archive/macros/latex/doc/fntguide.pdf](http://tex-archive/macros/latex/doc/fntguide.pdf)
- [9] D. P. Carlisle. *Packages in the ‘graphics’ bundle*. Je součástí kolekce „graphics“ – soubor `grfguide.tex` a získáte ho ze stejného zdroje

- jako svou L^AT_EXovou distribuci. CTAN:
[//tex-archive/macros/latex/required/graphics/grfguide.pdf](http://tex-archive/macros/latex/required/graphics/grfguide.pdf)
- [10] Rainer Schöpf, Bernd Raichle, Chris Rowley. *A New Implementation of L^AT_EX's Verbatim Environments*. Je součástí kolekce „tools“ – soubor `verbatim.dtx` a získáte ho ze stejného zdroje jako svou L^AT_EXovou distribuci. CTAN:
[//tex-archive/macros/latex/required/tools/verbatim.pdf](http://tex-archive/macros/latex/required/tools/verbatim.pdf)
- [11] Vladimir Volovich, Werner Lemberg and L^AT_EX3 Project Team. *Cyrillic languages support in L^AT_EX*. Je součástí distribuce L^AT_EX 2_ε – soubor `cyrguide.tex`.
CTAN:[//tex-archive/macros/latex/doc/cyrguide.pdf](http://tex-archive/macros/latex/doc/cyrguide.pdf)
- [12] Graham Williams. *The T_EX Catalogue* je téměř kompletním seznamem T_EXových a L^AT_EXových balíků. Dostupný online na
CTAN:[//tex-archive/](http://tex-archive/)
- [13] Keith Reckdahl. *Using EPS Graphics in L^AT_EX 2_ε Documents* vysvětluje vše týkající se souborů EPS a jejich použití v L^AT_EXových dokumentech. Dostupný online na
CTAN:[///tex-archive/info/epslatex/english/epslatex.pdf](http://tex-archive/info/epslatex/english/epslatex.pdf)
- [14] Kristoffer H. Rose. *X_Y-pic User's Guide*. Ke stažení na CTAN, spolu s distribucí X_Y-pic.
CTAN:[//tex-archive/macros/generic/diagrams/xypic/](http://tex-archive/macros/generic/diagrams/xypic/)
- [15] John D. Hobby. *A User's Manual for METAPOST*. Ke stažení na
<http://ect.bell-labs.com/who/hobby/MetaPost.html>
- [16] Alan Hoenig. *T_EX Unbound*. Oxford University Press, 1998. ISBN 0-19-509685-1. ISBN 0-19-509686-X (pbk.).
- [17] Urs Oswald. *Graphics in L^AT_EX 2_ε*, obsahující trochu Javových zdrojových kódů pro generování libovolných kružnic a elips v rámci prostředí `picture` a *METAPOST – A Tutorial*. Obojí ke stažení na
<http://www.ursoswald.ch/>
- [18] Till Tantau. *TikZ&PGF Manual*. Ke stažení na
CTAN:[//tex-archive/graphics/pgf/base/doc/generic/pgf/pgfmanual.pdf](http://tex-archive/graphics/pgf/base/doc/generic/pgf/pgfmanual.pdf)

Rejstřík

Symboly

..., 21
~, 33
. (mezera za .), 33
i a j bez tečky, 23
\!, 65, 66
", 19
"-, 32
"', 32
"<, 32
"=", 32
">, 32
"‘, 32
\$, 49
,', 54
\,, 52, 65
\-, 18, 20
\,, 17, 38, 39, 41, 126
*, 17
\:, 65
\;, 65
\@, 33
\[, 50, 51
\], 50, 51
^, 53
-, 53
l, 41

A

å, 23
abstract, 39
Acrobat Reader, 89
acute, 23
\addtolength, 128
æ, 23
aeguill, 90

akcent, 22
align, 59
\Alph, 31
\alph, 31
Americká matematická společnost,
49
amsbsy, 68
amsfonts, 67, 76
amsmath, 49, 55, 56, 65, 66, 68
amssymb, 52, 67, 72
amsthm, 68, 69
\and, 35
ansinew, 24
apostrof, 54
\appendix, 34, 35
applemac, 24
\arccos, 54
\arcsin, 54
\arctan, 54
\arg, 54
array, 64, 65
\arraystretch, 42
\Asbuk, 31
\asbuk, 31
\author, 35, 94

B

babel, 18, 22, 23, 30–32
\backmatter, 35
balík, 6, 11, 115
balík CJK, 28
balík makeidx, 84
balíky
aeguill, 90
amsbsy, 68
amsfonts, 67, 76

- amsmath, 49, 55, 56, 65, 66, 68
- amssymb, 52, 67, 72
- amsthm, 68, 69
- babel, 18, 22, 23, 30–32
- beamer, 96–98
- bicig, 32
- bm, 68
- calc, 128
- color, 96
- dcolumn, 42
- doc, 12
- eepic, 101, 105
- epic, 101
- eurosym, 21
- exscale, 12
- fancyhdr, 85, 86
- fontenc, 12, 24, 31
- geometry, 87
- graphicx, 81, 91, 96
- hyperref, 89, 92, 95, 96
- hyphenat, 87
- IEEEtrantools, 60
- ifpdf, 95
- ifthen, 12
- indentfirst, 124
- inputenc, 12, 24, 31
- latexsym, 12
- layout, 126
- longtable, 42
- makeidx, 12, 84
- mathrsfs, 76
- mathtext, 31
- mhchem, 67
- mltex, 90
- ntheorem, 69
- pgfplots, 113
- pgfplotstable, 113
- ppower4, 98
- prosper, 98
- pstricks, 101, 105
- pxfonts, 91
- showidx, 85
- syntonly, 12, 14
- textcomp, 20, 21
- tikz, 111
- txfonts, 91
- ucs, 24
- verbatim, 6, 87
- xalx, 31
- \bar, 54
- barevný text, 11
- beamer, 96–98
- \begin, 37, 102, 111, 117
- bezpatkové písmo, 120
- \bibitem, 83
- bibliografie, 83
- bicig, 32
- \Big, 57
- \big, 57
- \big(, 63
- \Bigg, 57
- \bigg, 57
- \bigskip, 126
- bílý znak, 4
 - na začátku řádku, 4
- binární relace, 56
- \binom, 55
- binomické koeficienty, 55
- Blackboard Bold, 52
- block, 99
- bm, 68
- Bmatrix, 65
- bmatrix, 65
- \bmod, 55
- \boldmath, 67
- \boldsymbol, 68
- brazilian, 25
- C**
- calc, 128
- \caption, 45, 46
- cases, 64
- \cdot, 53
- \cdots, 53
- center, 38
- \chapter, 34, 85
- \chaptermark, 85
- \ci, 115

- `\circle`, 105
- `\circle*`, 105
- `\cite`, 83
- `\cleardoublepage`, 46
- `\clearpage`, 46
- `\cline`, 41
 - color, 96
 - comment, 6
- `\cos`, 54
- `\cosh`, 54
- `\cot`, 54
- `\coth`, 54
 - cp850, 24
 - cp866nav, 24
 - cp1251, 24, 32
- `\csc`, 54

- Č
 - čárka, 21
 - čtverčiková pomlčka, 20

- D
 - `\date`, 35
 - dcolumn, 42
 - `\ddots`, 53
 - `\DeclareMathOperator`, 54
 - `\deg`, 54
 - `\depth`, 130, 131
 - derivace, 54
 - description, 37
 - `\det`, 54, 63
 - `\dfrac`, 55
 - diagonální tečky, 53
 - `\dim`, 54
 - dimensions, 124
 - displaymath, 50
 - `\displaystyle`, 67
 - dlouhé rovnice, 57
 - doc, 12
 - `\documentclass`, 9, 13, 18
 - double line spacing, 123
 - `\dq`, 27
 - `\dum`, 115

- E
 - eepic, 101, 105
 - `\emph`, 36, 120
 - empty, 11
 - Encapsulated POSTSCRIPT, 81, 91
 - `\end`, 37, 102, 117
 - `\enumBul`, 31
 - `\enumEng`, 31
 - enumerate, 37
 - `\enumLat`, 31
 - epic, 101
 - eqnarray, 59
 - `\eqref`, 50
 - equality, 59
 - equation, 50, 51, 57
 - amsmath, 50
 - L^AT_EX, 50
 - multiple, 58
 - equation*, 50, 51, 57
 - eurosym, 21
 - `\exp`, 54
 - exponent, 53
 - exscale, 12
 - extension
 - .aux, 13
 - .cls, 13
 - .dtx, 13
 - .dvi, 13, 82
 - .fd, 13
 - .idx, 13
 - .ilg, 14
 - .ind, 14, 85
 - .ins, 13
 - .lof, 13
 - .log, 13
 - .lot, 13
 - .sty, 13, 87
 - .tex, 8, 13
 - .toc, 13

- F
 - fancyhdr, 85, 86
 - `\fbox`, 19
 - figure, 44, 45

- `\flq`, 27
 - `\flqq`, 27
 - `flushleft`, 38
 - `flushright`, 38
 - `\foldera`, 109
 - `\folderb`, 109
 - font, 119
 - `\footnotesize`, 120
 - `\Huge`, 120
 - `\huge`, 120
 - `\LARGE`, 120
 - `\Large`, 120
 - `\large`, 120
 - `\mathbf`, 121
 - `\mathcal`, 121
 - `\mathit`, 121
 - `\mathnormal`, 121
 - `\mathrm`, 121
 - `\mathsf`, 121
 - `\mathtt`, 121
 - `\normalsize`, 120
 - `\scriptsize`, 120
 - `\small`, 120
 - `\textbf`, 120
 - `\textit`, 120
 - `\textmd`, 120
 - `\textnormal`, 120
 - `\textrm`, 120
 - `\textsc`, 120
 - `\textsf`, 120
 - `\textsl`, 120
 - `\texttt`, 120
 - `\textup`, 120
 - `\tiny`, 120
 - font – kódování
 - LGR, 25
 - OT1, 24
 - T1, 25, 31
 - T2*, 30
 - T2A, 25, 31
 - T2B, 25
 - T2C, 25
 - X2, 25
 - font – velikost, 119
 - fontenc, 12, 24, 31
 - fonty – kódování, 24
 - `\footnote`, 36, 46
 - `\footnotesize`, 120
 - `\footskip`, 127
 - `\frac`, 55
 - frame, 98
 - `\framebox`, 130
 - francouzština, 26
 - `\frenchspacing`, 31, 33
 - `\frontmatter`, 35
 - `\frq`, 27
 - `\frqq`, 27
 - funkce modulo, 55
 - `\fussy`, 18
- ## G
- `\gcd`, 54
 - geometry, 87
 - GhostScript, 81
 - GhostView, 135
 - Gimp, 135
 - grafika, 11, 81
 - graphicx, 81, 91, 96
 - grave, 23
 - Greek, 30
 - grouping, 120
- ## H
- `\hat`, 54
 - `\headheight`, 127
 - `textttheadings`, 11
 - `\headsep`, 127
 - `\height`, 130, 131
 - HL^AT_EX, 28
 - hL^AT_EXp, 28
 - `\hline`, 41
 - `\hom`, 54
 - horizontal
 - space, 124
 - horizontální
 - linka, 53
 - svorka, 53
 - tečky, 53

- hranaté závorky, 5
- `\href`, 93, 94, 96
- `\hspace`, 117, 124
- `\Huge`, 120
- `\huge`, 120
- hyperref, 89, 92, 95, 96
- hypertextový, 88
- hyphenat, 87
- `\hyphenation`, 18
- I**
- `\idotsint`, 66
- `\IEEEeqnarray`, 58–61
- `\IEEEeqnarraymulticol`, 62
- `\IEEEmulticol`, 63
- `\IEEEnonumber`, 63
- `\IEEEtrantools`, 60
- `\IEEEyesnumber`, 63
- `\IEEEyessubnumber`, 64
- ifpdf, 95
- ifthen, 12
- `\ignorespaces`, 117
- `\ignorespacesafterend`, 117
- `\iiiint`, 66
- `\iiint`, 66
- `\iint`, 66
- `\include`, 14
- `\includegraphics`, 82, 91, 95, 129
- `\includeonly`, 14
- `\indent`, 124
- indentfirst, 124
- `\index`, 53, 84, 85
- `\inf`, 54
- `\input`, 14, 118
- inputenc, 12, 24, 31
- `\int`, 56, 63
- internacionalizace, 22
- italika, 120
- `\item`, 37
- itemize, 37
- J**
- jazyk, 22
- jednostranný, 10
- jednotky, 124
- K**
- kapitálky, 120
- `\ker`, 54
- Knuth, Donald E., 1
- kódování
- fontů
- LGR, 25
- OT1, 24
- T1, 25, 31
- T2*, 30
- T2A, 25, 31
- T2B, 25
- T2C, 25
- X2, 25
- vstupní
- ansinew, 24
- applemac, 24
- cp850, 24
- cp866nav, 24
- cp1251, 24, 32
- koi8-ru, 24, 31
- latin1, 24
- macukr, 24
- mn, 32
- utf8, 32
- utf8x, 24
- kódování fontů, 12
- koi8-ru, 24, 31
- komentáře, 6
- korejský font
- font UHC, 29
- korejština, 27
- křížové odkazy, 35
- L**
- `\label`, 35, 36, 46, 50
- Lamport, Leslie, 1
- `\LARGE`, 120
- `\Large`, 120
- `\large`, 120
- `\LaTeX`, 19
- L^AT_EX3, 4

- `\LaTeXe`, 19
- `latexsym`, 12
- `latin1`, 24
- `layout`, 126
- `\ldots`, 21, 53
- `\left`, 56, 63
- `\left.`, 56
- `\lefteqn`, 60, 62
- `\leftmark`, 85
- `\lg`, 54
 - LGR, 25
 - ligatura, 22
- `\lim`, 54
- `\liminf`, 54
- `\limsup`, 54
- `\line`, 104, 109
- `\linebreak`, 17
- `\linespread`, 123
- `\linethickness`, 106, 107, 109
 - linka
 - horizontální, 53
- `\listoffigures`, 45
- `\listoftables`, 45
- `\ln`, 54
- `\log`, 54, 63
 - longtable, 42
 - lscmmand, 115
- M**
- MacTeX, 134
- macukr, 24
- `\mainmatter`, 35, 94
- `\makebox`, 130
 - makeidx, 12, 84
- `\makeindex`, 84
 - makeindex – program, 84
- `\maketitle`, 35
- `\marginparpush`, 127
- `\marginparsep`, 127
- `\marginparwidth`, 127
- matematické
 - akcenty, 54
 - delimitery, 57
 - funkce, 54
 - mínus, 20
 - matematické mezery, 65
 - matematický mód, 51
 - matematický znak mínus, 20
 - matematika, 49
- `\mathbb`, 52
- `\mathbf`, 121
- `\mathcal`, 121
- `\mathit`, 121
- `\mathnormal`, 121
- `\mathrm`, 121
 - mathrsfs, 76
- `\mathsf`, 121
 - mathtext, 31
- `\mathtt`, 121
- matrix, 65
- `\max`, 54, 63
- `\mbox`, 19, 22, 130
- `\medskip`, 126
- METAPOST, 91, 92
- mezera, 4
- mezery
 - v matemetickém módu, 51
 - za příkazy, 5
- mhchem, 67
- MiKTeX, 134
- `\min`, 54
 - minimální třída, 9
- minipage, 130
- Mittelbach, Frank, 1
- mltex, 90
- mn, 32
- `\multicolumn`, 42
- `\multicolumns`, 62
- `\multipt`, 103, 106, 107
- multline, 57–59
- multline*, 57
- N**
- němčina, 23, 27
- nepovinný parametr, 5
- `\newcommand`, 66, 116
- `\newenvironment`, 117
- `\newline`, 17

- `\newpage`, 17
- `\newsavebox`, 108
- `\newtheorem`, 68
- `\noindent`, 124
- `\nolinebreak`, 17
- `\nonumber`, 63
- `\nopagebreak`, 17
- `\normalsize`, 120
- `\not`, 73
- `ntheorem`, 69

- O**
- oboustranný, 10
- oddělovač, 56
- `\oddsidemargin`, 127
- odmocnina, 53
- odstavec, 15
- œ, 23
- OT1, 24
- `\oval`, 107, 109
- `\overbrace`, 53
- `\overleftarrow`, 54
- `\overline`, 53
- `\overrightarrow`, 54

- P**
- `\pagebreak`, 17
- `\pageref`, 35, 88
- `\pagestyle`, 11
- paper size, 89
- `\paperheight`, 127
- `\paperwidth`, 127
- `\par`, 122
- `\paragraph`, 34
- parametr, 5
- `\parbox`, 130
- parciální derivace, 55
- `\parindent`, 124
- `\parskip`, 124
- `\part`, 34
- `\partial`, 55
- pata stránky, 11
- PDF, 88
- pdfL^AT_EX, 89, 96
- pdfT_EX, 89
- PDFView, 134
- pgfplots, 113
- pgfplotstable, 113
- `\phantom`, 46, 66
- picture, 101, 102, 105, 106
- plain, 11
- pmatrix, 65
- `\pmod`, 55
- Português, 25
- portuguese, 25
- POSTSCRIPT, 29, 46, 89, 90, 101, 102
- Encapsulated, 81, 91
- povolenka umístění, 44
- ppower4, 98
- `\Pr`, 54
- preamble, 6
- `\printindex`, 85
- `\prod`, 56
- proof, 69
- prosper, 98
- prostředí
 - abstract, 39
 - align, 59
 - array, 64, 65
 - block, 99
 - Bmatrix, 65
 - bmatrix, 65
 - cases, 64
 - center, 38
 - comment, 6
 - description, 37
 - displaymath, 50
 - enumerate, 37
 - eqnarray, 59
 - equality, 59
 - equation, 50, 51, 57
 - equation*, 50, 51, 57
 - figure, 44, 45
 - flushleft, 38
 - flushright, 38
 - frame, 98
 - IEEEeqnarray, 58–61

- itemize, 37
- lscommand, 115
- matrix, 65
- minipage, 130
- multline, 57–59
- multline*, 57
- parbox, 130
- picture, 101, 102, 105, 106
- pmatrix, 65
- proof, 69
- pspicture, 101
- quotation, 39
- quote, 39
- table, 44, 45
- tabular, 40, 129
- thebibliography, 83
- tikzpicture, 111
- verbatim, 40, 87
- verse, 39
- Vmatrix, 65
- vmatrix, 65
- \protect, 46, 47
- \providecommand, 116
- \ProvidesPackage, 119
- příkaz, 5
- příkazy
 - \!, 65, 66
 - \., 52, 65
 - \-, 18
 - \, 17, 38, 39, 41, 126
 - *, 17
 - \:, 65
 - \;, 65
 - \@, 33
 - \[, 50, 51
 - \], 50, 51
 - \addtolength, 128
 - \Alph, 31
 - \alph, 31
 - \and, 35
 - \appendix, 34, 35
 - \arccos, 54
 - \arcsin, 54
 - \arctan, 54
 - \arg, 54
 - \arraystretch, 42
 - \Asbuk, 31
 - \asbuk, 31
 - \author, 35, 94
 - \backmatter, 35
 - \bar, 54
 - \begin, 37, 102, 111, 117
 - \bibitem, 83
 - \Big, 57
 - \big, 57
 - \big(, 63
 - \Bigg, 57
 - \bigg, 57
 - \bigskip, 126
 - \binom, 55
 - \bmod, 55
 - \boldmath, 67
 - \boldsymbol, 68
 - \caption, 45, 46
 - \cdot, 53
 - \cdots, 53
 - \chapter, 34, 85
 - \chaptermark, 85
 - \ci, 115
 - \circle, 105
 - \circle*, 105
 - \cite, 83
 - \cleardoublepage, 46
 - \clearpage, 46
 - \cline, 41
 - \cos, 54
 - \cosh, 54
 - \cot, 54
 - \coth, 54
 - \csc, 54
 - \date, 35
 - \ddots, 53
 - \DeclareMathOperator, 54
 - \deg, 54
 - \depth, 130, 131
 - \det, 54, 63
 - \dfrac, 55
 - \dim, 54

`\displaystyle`, 67
`\documentclass`, 9, 13, 18
`\dq`, 27
`\dum`, 115
`\emph`, 36, 120
`\end`, 37, 102, 117
`\enumBul`, 31
`\enumEng`, 31
`\enumLat`, 31
`\eqref`, 50
`\exp`, 54
`\fbox`, 19
`\flq`, 27
`\flqq`, 27
`\foldera`, 109
`\folderb`, 109
`\footnote`, 36, 46
`\footskip`, 127
`\frac`, 55
`\framebox`, 130
`\frenchspacing`, 31, 33
`\frontmatter`, 35
`\frq`, 27
`\frqq`, 27
`\fussy`, 18
`\gcd`, 54
`\hat`, 54
`\headheight`, 127
`\headsep`, 127
`\height`, 130, 131
`\hline`, 41
`\hom`, 54
`\href`, 93, 94, 96
`\hspace`, 117, 124
`\hyphenation`, 18
`\idotsint`, 66
`\IEEEeqnarraymulticol`, 62
`\IEEEmulticol`, 63
`\IEEEnonumber`, 63
`\IEEEyesnumber`, 63
`\IEEEyessubnumber`, 64
`\ifpdf`, 95
`\ignorespaces`, 117
`\ignorespacesafterend`, 117
`\iiiint`, 66
`\iiint`, 66
`\iint`, 66
`\include`, 14
`\includegraphics`, 82, 91, 95, 129
`\includeonly`, 14
`\indent`, 124
`\index`, 84, 85
`\inf`, 54
`\input`, 14, 118
`\int`, 56, 63
`\item`, 37
`\ker`, 54
`\label`, 35, 36, 46, 50
`\LaTeX`, 19
`\LaTeXe`, 19
`\ldots`, 21, 53
`\left`, 56, 63
`\left.`, 56
`\lefteqn`, 60, 62
`\leftmark`, 85
`\lg`, 54
`\lim`, 54
`\liminf`, 54
`\limsup`, 54
`\line`, 104, 109
`\linebreak`, 17
`\linespread`, 123
`\linethickness`, 106, 107, 109
`\listoffigures`, 45
`\listoftables`, 45
`\ln`, 54
`\log`, 54, 63
`\mainmatter`, 35, 94
`\makebox`, 130
`\makeindex`, 84
`\maketitle`, 35
`\marginparpush`, 127
`\marginparsep`, 127
`\marginparwidth`, 127
`\mathbb`, 52
`\max`, 54, 63
`\mbox`, 19, 22, 130

<code>\medskip</code> , 126	<code>\qedhere</code> , 69, 70
<code>\min</code> , 54	<code>\qqquad</code> , 52, 65
<code>\multicolumn</code> , 42	<code>\quad</code> , 52, 62, 65
<code>\multicolumns</code> , 62	<code>\raisebox</code> , 131
<code>\multipt</code> , 103, 106, 107	<code>\ref</code> , 35, 46, 88
<code>\newcommand</code> , 66, 116	<code>\renewcommand</code> , 116
<code>\newenvironment</code> , 117	<code>\renewenvironment</code> , 117
<code>\newline</code> , 17	<code>\right</code> , 56, 63
<code>\newpage</code> , 17	<code>\rightmark</code> , 85
<code>\newsavebox</code> , 108	<code>\rule</code> , 117, 131
<code>\newtheorem</code> , 68	<code>\savebox</code> , 108
<code>\noindent</code> , 124	<code>\scriptscriptstyle</code> , 67
<code>\nolinebreak</code> , 17	<code>\scriptstyle</code> , 67
<code>\nonumber</code> , 63	<code>\sec</code> , 54
<code>\nopagebreak</code> , 17	<code>\section</code> , 34, 46, 85
<code>\not</code> , 73	<code>\sectionmark</code> , 85
<code>\oddsidemargin</code> , 127	<code>\selectlanguage</code> , 23
<code>\oval</code> , 107, 109	<code>\setlength</code> , 102, 124, 128
<code>\overbrace</code> , 53	<code>\settodepth</code> , 128
<code>\overleftarrow</code> , 54	<code>\settoheight</code> , 128
<code>\overline</code> , 53	<code>\settowidth</code> , 128
<code>\overrightarrow</code> , 54	<code>\sin</code> , 54, 63
<code>\pagebreak</code> , 17	<code>\sinh</code> , 54
<code>\pageref</code> , 35, 88	<code>\sloppy</code> , 18
<code>\pagestyle</code> , 11	<code>\smallskip</code> , 126
<code>\paperheight</code> , 127	<code>\smash</code> , 51
<code>\paperwidth</code> , 127	<code>\sqrt</code> , 53
<code>\par</code> , 122	<code>\stackrel</code> , 56
<code>\paragraph</code> , 34	<code>\stretch</code> , 117, 125
<code>\parbox</code> , 130	<code>\subparagraph</code> , 34
<code>\parindent</code> , 124	<code>\subsection</code> , 34
<code>\parskip</code> , 124	<code>\subsectionmark</code> , 85
<code>\part</code> , 34	<code>\substack</code> , 56
<code>\partial</code> , 55	<code>\subsubsection</code> , 34
<code>\phantom</code> , 46, 66	<code>\sum</code> , 56, 63
<code>\pmod</code> , 55	<code>\sup</code> , 54
<code>\Pr</code> , 54	<code>\tabcolsep</code> , 42
<code>\printindex</code> , 85	<code>\tableofcontents</code> , 34
<code>\prod</code> , 56	<code>\tag</code> , 50
<code>\protect</code> , 46, 47	<code>\tan</code> , 54
<code>\providecommand</code> , 116	<code>\tanh</code> , 54
<code>\ProvidesPackage</code> , 119	<code>\TeX</code> , 19
<code>\put</code> , 103–108	<code>\texorpdfstring</code> , 95
<code>\qbezier</code> , 101, 103, 110	<code>\textbackslash</code> , 5

- `\textcelsius`, 20
 - `\textdegree`, 20
 - `\texteuro`, 21
 - `\textheight`, 127
 - `\textstyle`, 67
 - `\textwidth`, 127, 129
 - `\tfrac`, 55
 - `\theoremstyle`, 68
 - `\thicklines`, 104, 107, 109
 - `\thinlines`, 107, 109
 - `\thispagestyle`, 11
 - `\title`, 35
 - `\tnss`, 116
 - `\today`, 19
 - `\topmargin`, 127
 - `\totalheight`, 130, 131
 - `\ud`, 66
 - `\underbrace`, 53
 - `\underline`, 36, 53
 - `\unitlength`, 102, 104
 - `\usebox`, 108
 - `\usepackage`, 11, 13, 21, 23, 24, 31, 119
 - `\usetikzlibrary`, 112
 - `\vdots`, 53
 - `\vec`, 54
 - `\vector`, 104
 - `\verb`, 40
 - `\verbatiminput`, 87
 - `\vspace`, 125
 - `\widehat`, 54
 - `\widetilde`, 54
 - `\width`, 130, 131
 - `pspicture`, 101
 - `pstricks`, 101, 105
 - půlčtverčiková pomlčka, 20
 - `\put`, 103–108
 - `pxfonts`, 91
- Q**
- `\qBezier`, 101, 103, 110
 - `\qedhere`, 69, 70
 - `\qqquad`, 52, 65
 - `\quad`, 52, 62, 65
 - quotation, 39
 - quote, 39
- R**
- `\raisebox`, 131
 - `\ref`, 35, 46, 88
 - `\renewcommand`, 116
 - `\renewenvironment`, 117
 - rezervovaný znak, 4
 - `\right`, 56, 63
 - `\rightmark`, 85
 - rozložení stránky, 126
 - `\rule`, 117, 131
- Ř**
- řádkové mezery, 123
 - řádkový zlom, 17
 - řecká písmena, 52
 - řečtina, 30
- S**
- `\savebox`, 108
 - `\scriptscriptstyle`, 67
 - `\scriptsize`, 120
 - `\scriptstyle`, 67
 - `\sec`, 54
 - `\section`, 34, 46, 85
 - `\sectionmark`, 85
 - `\selectlanguage`, 23
 - `\setlength`, 102, 124, 128
 - `\settodepth`, 128
 - `\settoheight`, 128
 - `\settowidth`, 128
 - showidx, 85
 - `\sin`, 54, 63
 - `\sinh`, 54
 - skandinávská písmena, 23
 - skloněné, 120
 - `\sloppy`, 18
 - sloupec – jeden, 10
 - sloupec – více sloupců, 10
 - slovo, 85
 - složené závorky, 5, 120
 - `\small`, 120
 - `\smallskip`, 126

- `\smash`, 51
 - speciální znak, 22
 - spojovník, 20
- `\sqrt`, 53
- `\stackrel`, 56
- stránkové styly, 11
- stránkový styl
 - empty, 11
 - headings, 11
 - plain, 11
- `\stretch`, 117, 125
- struktura vstupního souboru, 6
- strut, 43
- stupeň, 20
- styl display, 49, 51
- styl text, 49, 51
- `\subparagraph`, 34
- `\subsection`, 34
- `\subsectionmark`, 85
- `\substack`, 56
- `\subsubsection`, 34
- `\sum`, 56, 63
- `\sup`, 54
- svorka, 56
 - horizontální, 53
- symbol integrálu, 56
- symbol součinu, 56
- symbol sumy, 56
- syntonly, 12, 14

- Š
- šipka, 54

- T
- T1, 25, 31
- T2*, 30
- T2A, 25, 31
- T2B, 25
- T2C, 25
- `\tabcolsep`, 42
- table, 44, 45
- `\tableofcontents`, 34
- tabular, 40, 129
- tabulka, 40

- `\tag`, 50
- `\tan`, 54
- `\tanh`, 54
 - tečka, 21
 - tečky
 - tři, 53
- `\TeX`, 19
 - TeXnicCenter, 135
- `\texorpdfstring`, 95
- `\textbackslash`, 5
- `\textbf`, 120
- `\textcelsius`, 20
- textcomp, 20, 21
- `\textdegree`, 20
- `\texteuro`, 21
- `\textheight`, 127
- `\textit`, 120
 - TextMate, 134
- `\textmd`, 120
- `\textnormal`, 120
 - textový mód, 51
- `\textrm`, 120
- `\textsc`, 120
- `\textsf`, 120
- `\textsl`, 120
- `\textstyle`, 67
- `\texttt`, 120
- `\textup`, 120
- `\textwidth`, 127, 129
- `\frac`, 55
 - thebibliography, 83
- `\theoremstyle`, 68
- `\thicklines`, 104, 107, 109
- `\thinlines`, 107, 109
- `\thispagestyle`, 11
- tikz, 111
- tikzpicture, 111
- tilda (~), 20, 33, 54
- `\tiny`, 120
- `\title`, 35
 - titul, 35
 - titul dokumentu, 10
 - titulní stránka, 10
- `\tnss`, 116

- `\today`, 19
- `\topmargin`, 127
- `\totalheight`, 130, 131
- třída article, 9
- třída book, 9
- třída proc, 9
- třída report, 9
- třída slides, 9
- tučné písmo, 120
- tučné symboly, 52, 67
- `txfonts`, 91
- typ souboru, 11

- U**
- ucs, 24
- `\ud`, 66
- umlaut, 23
- `\underbrace`, 53
- `\underline`, 36, 53
- `\unitlength`, 102, 104
- units, 125
- URL, 20
- úřední dokument, 10
- `\usebox`, 108
- `\usepackage`, 11, 13, 21, 23, 24, 31, 119
- `\usetikzlibrary`, 112
- utf8, 32
- utf8x, 24
- uvozovky, 19

- V**
- varování overfull hbox, 17
- varování underfull hbox, 18
- `\vdots`, 53
- `\vec`, 54
- `\vector`, 104
- vektory, 54
- velikost A4, 10
- velikost A5, 10
- velikost B5, 10
- velikost „dopis“, 10
- velikost exekutiva, 10
- velikost fontu, 120
- velikost fontu dokumentu, 10
- velikost papíru, 10, 126
- velikost základního fontu, 10
- `\verb`, 40
- verbatim, 6, 40, 87
- `\verbatiminput`, 87
- verse, 39
- vertical space, 126
- vertikální
 - tečky, 53
- `Vmatrix`, 65
- `vmatrix`, 65
- `\vspace`, 125
- vstupní kódování
 - ansinew, 24
 - applemac, 24
 - cp850, 24
 - cp866nav, 24
 - cp1251, 24, 32
 - koi8-ru, 24, 31
 - latin1, 24
 - macukr, 24
 - mn, 32
 - utf8, 32
 - utf8x, 24
- vstupní soubor, 7
- vstupní soubory v korejštině, 28
- výpustka, 21
- vzorce, 49
- vzpřímené, 120
- vzpřímené písmo, 120

- W**
- `\widehat`, 54
- `\widetilde`, 54
- `\width`, 130, 131
- www, 20
- WYSIWYG, 2, 3

- X**
- X2, 25
- xalx, 31
- Xpdf, 89

Z

- záhlaví stránky, 11
- zarovnané doleva, 38
- zarovnání desetinných čísel, 42
- zlomek, 55
- zpětné lomítko, 5
- zranitelné příkazy, 46