

Nie za krótkie wprowadzenie do systemu $\text{\LaTeX} 2_{\epsilon}$

Albo $\text{\LaTeX} 2_{\epsilon}$ w 129 minut

Tobias Oetiker

Hubert Partl, Irene Hyna, Elisabeth Schlegl

Tomasz Przechlewski i Ryszard Kubiak

Janusz Gołdasz

Wydanie drugie, poprawione, uaktualnione i rozszerzone
Oparte na wersji 4.20 *The Not So Short Introduction to $\text{\LaTeX} 2_{\epsilon}$*
z 31 maja 2006

Styczeń 2007

Podziękowania

Większość materiału w niniejszej książce pochodzi z napisanego w języku niemieckim austriackiego *Wprowadzenia do L^AT_EX₂ε*, którego autorami są:

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

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

Elisabeth Schlegl <no email>
in Graz

Osoby zainteresowane wersją niemiecką, opracowaną przez Jörga Knappena do L^AT_EX₂ε, mogą ją znaleźć pod adresem CTAN://info/lshort/german.

Niżej wymienione osoby pomogły w tworzeniu *Wprowadzenia* swoimi poprawkami, sugestiami i propozycjami ulepszeń. Przyczyniły się one bardzo do nadania niniejszej książce jej obecnego kształtu. Chciałbym im za to serdecznie podziękować. Za wszelkie błędy, które nieuchronnie pozostały w tekście, ponoszę oczywiście odpowiedzialność wyłącznie ja sam. Wszystkie natomiast słowa zapisane bez błędu są w książce wyłączną zasługą osób z poniższej listy:

Rosemary Bailey, Marc Bevand, Friedemann Brauer, Jan Busa,
Markus Brühwiler, Pietro Braione, David Carlisle, José Carlos Santos,
Neil Carter, Mike Chapman, Pierre Chardaire, Christopher Chin, Carl Cerecke,
Chris McCormack, Wim van Dam, Jan Dittberner, Michael John Downes,
Matthias Dreier, David Dureisseix, Elliot, Hans Ehrbar, Daniel Flipo, David Frey,
Hans Fugal, Robin Fairbairns, Jörg Fischer, Erik Frisk, Mic Milic Frederickx,
Frank, Kasper B. Graversen, Arlo Griffiths, Alexandre Guimond, Andy Goth,
Cyril Goutte, Greg Gamble, Frank Fischli, Morten Hgholm, Neil Hammond,
Rasmus Borup Hansen, Joseph Hilferty, Björn Hvittfeldt, Martien Hulsen,
Werner Icking, Jakob, Eric Jacoboni, Alan Jeffrey, Byron Jones, David Jones,
Johannes-Maria Kaltenbach, Michael Koundouros, Andrzej Kawalec,
Sander de Kievit, Alain Kessi, Christian Kern, Tobias Klauser, Jörg Knappen,
Kjetil Kjernsmo, Maik Lehradt, Rémi Letot, Flori Lambrechts, Axel Liljencrantz,
Johan Lundberg, Alexander Mai, Hendrik Maryns, Martin Maechler,
Aleksandar S Milosevic, Henrik Mitsch, Claus Malten, Kevin Van Maren,

- OpenType, 28
- operator
 - ▷ iloczynu, 50
 - ▷ sumowania, 50
- OT1*, 27, 29
- OT4*, 27, 29
- otoczenie, 35
- \oval, 90, 92
- \overbrace, 48
- overfull hbox, 19
- \overleftarrow, 48
- \overline, 48
- \overrightarrow, 48
- \pagebreak, 17, 18, 110
- \pageref, 34, 74
- \pagestyle, 11
 - pakiet, 10
- \paperheight, 112
- \paperwidth, 112
- \par, 104, 107
- \paragraph, 31
- \parbox, 114
- \parindent, 108
- \parskip, 108
- \part, 31, 32
- \pause, 83
- .pdf (plik), 13, 67, 73, 76, 80
- PDF, 74, 75, 78
- pdfL^AT_EX, 75, 76, 81
- pdfscreen, 83
- pdfTeX*, 67, 76, 77
- pdfT_EX, 75
- \phantom, 54
- Pianowski Piotr, 122
- picture, 84, 85, 88, 89
- pierwiastek kwadratowy, 48
- pismo
 - ▷ drukarskie, 102
 - ▷ krój, 102
 - ▷ odmiana, 102
 - ▷ stopień, 102
- platex, 22, 28, 29
- plmath*, 29
- plmindex, 13, 70
- \pmb, 57
- \pmod, 49
- .png (plik), 67, 76, 80
- podpis, 68
- polecenie, 5
 - ▷ kruche, 43
- \polecenie, 98
- polski, 29–31, 49, 59, 95
- PostScript, 9, 42, 75, 85
- ppower4, 81
- preambula, 6
- prim, 48
- \printindex, 70
- \prod, 50
- proof, 57
- prosper, 81
- \protect, 43
- \providecommand, 100
- \ProvidesPackage, 101
 - przecinek, 24
- .ps (plik), 73
- \ps, 44
- pspicture, 85
- pstricks, 75, 85, 88
- pudełko, 68, 113
- \put, 86–91
- pxfonts, 106
- \qbezier, 84, 86, 93
- qcourier, 106
- \qedhere, 57
- qpalatin, 106
- qpxmath, 106
- \qqquad, 52, 110
- qswiss, 106
- qtimes, 106
- qtxmath, 106
- \quad, 46, 52, 110
- quotation, 37
- quote, 37
- Radhakrishnana C.V., 83
- \raisebox, 115
- \ref, 34, 42, 46, 68, 74
- \renewcommand, 100
- \renewenvironment, 100
- \right, 50, 51
- \rightmark, 72
- rotate, 75
- rotate, 68
- rotating, 68
- \rule, 115, 116
- Ryćko Marek, 20, 27
- \savebox, 91
- \scriptscriptstyle, 55
- \scriptsize, 103
- \scriptstyle, 55
- \section, 31, 43, 72, 83
- \sectionmark, 72
- \selecthyphenation, 30
- \selectlanguage, 30
- \setlength, 85, 108, 111
- \settodepth, 113
- \settoheight, 113
- \settowidth, 113
- sheaders, 106
- showidx, 70
- sideways, 68
- sidewaysfigure, 68
- sidewaystable, 68
- \signature, 43, 44
- Simpson Tom, 120
- \sloppy, 19
- slowo, 71
- \small, 103
- \smallskip, 110
- split, 54
- spójka, 24
- \sqrt, 48
- \stackrel, 50
- stopień pisma, 103
- \stretch, 109, 110, 117
- Strzelczyk Piotr, 122
- .sty (plik), 12, 73, 101
- subarray, 50
- \subparagraph, 31
- \subsection, 31, 83
- \subsectionmark, 72
- \substack, 50
- \subsubsection, 31
- \sum, 50
- \surd, 48
- symbol
 - ▷ końca dowodu, 57
- syntony, 12
- środowisko, *zob.* otoczenie
- T1*, 27, 29
- T2A*, 28
- T2B*, 28
- T2C*, 28
- tabbing, 28
- \tabcolsep, 117
- table, 40–42
- \tableofcontents, 32, 42

Przedmowa

L^AT_EX [12] jest systemem składu znakomicie nadającym się do tworzenia publikacji naukowych i technicznych o wysokiej jakości typograficznej. L^AT_EX nadaje się również do przygotowywania dowolnego rodzaju dokumentów, poczynając od prostych listów, a kończąc na grubych książkach. Do formatowania dokumentów L^AT_EX wykorzystuje program T_EX [11, 5].

Niniejsze krótkie *Wprowadzenie* opisuje L^AT_EXa w zakresie wystarczającym do większości zastosowań. Pełny opis L^AT_EXa można znaleźć w [12, 6, 4].

Wprowadzenie zawiera sześć następujących rozdziałów:

Rozdział 1 przedstawia ogólną strukturę dokumentów L^AT_EXowych i krótko omawia historię systemu L^AT_EX. Po przeczytaniu tego rozdziału powinien mieć już zgrubne wyobrażenie o tym, czym jest L^AT_EX.

Rozdział 2 podaje szczegóły dotyczące składania dokumentów. Omówiono w nim najważniejsze instrukcje i otoczenia. Po przeczytaniu tego rozdziału będziesz już umiał tworzyć proste dokumenty L^AT_EXowe.

Rozdział 3 poświęcono składaniu wzorów matematycznych. Wiele zamieszczonych w nim przykładów nauczy Cię wykorzystywać jedną z najważniejszych umiejętności T_EXa, jaką jest elegancki skład matematyki. Na końcu rozdziału zamieszczono zestawienie dostępnych w L^AT_EXu symboli matematycznych.

Rozdział 4 wyjaśnia, jak tworzyć skorowidze i spisy bibliograficzne oraz jak dołączać rysunki w formacie EPS. Znajdziesz tu również informacje o tworzeniu plików w formacie PDF za pomocą programu *pdflatex* oraz o kilku użytecznych pakietach L^AT_EXowych.

Rozdział 5 pokazuje, jak używać L^AT_EXa do tworzenia grafiki. Zamiast przygotowywać rysunek w jakimś programie graficznym, zachować go w pliku, po czym włączyć do dokumentu L^AT_EXowego, opisujesz rysunek w dokumencie, a do jego narysowania używasz samego L^AT_EXa.

Rozdział 6 zawiera informacje, których wykorzystanie jest potencjalnie niebezpieczne, bo mówią o tym, jak można zmienić standardowy układ graficzny dokumentów L^AT_EXowych. Niewłaściwe ich użycie może spowodować pogorszenie ładnego na ogół składu L^AT_EX-owego.

Sądzymy, że powinieneś przeczytać wszystkie rozdziały, w powyższej kolejności. Ostatecznie, książka nie jest zbyt gruba. Szczególną uwagę zwróć na przykłady, gdyż właśnie w nich zawarto sporo wartościowych informacji.

<code>\cline</code> , 40, 118	<code>\fbox</code> , 20	<code>\height</code> , 114, 115
<code>\closing</code> , 44	<code>.fd</code> (plik), 13	hiperłącze, 74
<code>.cls</code> (plik), 13	<code>figure</code> , 40–42, 67	hipertekst, 74
<code>cmd</code> , 8	<code>flalign</code> , 54	<code>\hline</code> , 39
<code>color</code> , 76, 81	<code>flegn</code> , 10	Hobby John D., 85
<code>colortbl</code> , 122	<code>flushleft</code> , 36	<code>\href</code> , 78, 80
<code>command</code> , 8	<code>flushright</code> , 36	<code>\hspace</code> , 108
<code>CorelDraw!</code> , 65	<code>foiltex</code> , 9	<code>\hspace*</code> , 109
<code>cp1250</code> , 8, 26	<code>\foldera</code> , 92	HTML, 74
czcionka, <i>zob.</i> pismo	<code>\folderb</code> , 92	<code>\huge</code> , 103
drukarskie	font, 102	<code>\Huge</code> , 103
<code>\date</code> , 33, 43, 44	fontenc, 12, 27, 29	hyperref, 77, 79–81
<code>dcolumn</code> , 39	fonty	hyphenat, 72
<code>\ddots</code> , 51	▷ CM, 27	<code>\hyphenation</code> , 19, 20
<code>\depth</code> , 114, 115	▷ cm-super, 76	ı („i” bez kropki), 25
description, 36	▷ EC, 27, 76	IDE, 75
<code>displaymath</code> , 45	▷ LM, 27, 76	<code>\idotsint</code> , 52
<code>\displaystyle</code> , 55	▷ PL, 27, 76	<code>.idx</code> (plik), 13, 70, 73
doc, 12	<code>\footnote</code> , 34, 43	<code>ifpdf</code> , 80
<code>\documentclass</code> , 6, 9, 10,	<code>\footnotesize</code> , 103	<code>\ifpdf</code> , 80
13, 19, 29, 81	<code>\footskip</code> , 112	<code>ifthen</code> , 12
<i>draft</i> , 19	format, 25, 26, 28	<code>\ignorespaces</code> , 101
<code>.dtx</code> (plik), 13, 73	formatowanie	<code>\ignorespacesafterend</code> ,
<code>.dvi</code> (plik), 9, 13, 66, 73	▷ logiczne, 2	101
<code>dvipdf</code> , 9	▷ wizualne, 2	<code>\iiiint</code> , 52
<i>dvips</i> , 67, 77	<code>\frac</code> , 49	<code>\iiint</code> , 52
<code>dvips</code> , 9, 13, 66, 74	<code>frame</code> , 83	<code>\iint</code> , 52
<code>\dywiz</code> , 22	<code>\framebox</code> , 115	<code>.ilg</code> (plik), 13
eeptic, 85, 88	<code>\frametitle</code> , 83	<code>ImageMagick</code> , 66
<code>\em</code> , 35	<code>Freehand</code> , 65	<code>\include</code> , 14, 67
<code>\emph</code> , 35, 103	<code>\frenchspacing</code> , 31	<code>\includegraphics</code> , 66, 67,
<code>enumerate</code> , 36	<code>\frontmatter</code> , 33	76, 80, 113
<code>epic</code> , 85	<code>\fussy</code> , 19	<code>\includeonly</code> , 14
<code>.eps</code> (plik), 76, 80	<code>gather</code> , 54	<code>.ind</code> (plik), 13, 70
EPS, 65, 66, 76	<code>geometry</code> , 72, 111	indeks
<code>epstopdf</code> , 77	<code>\geq</code> , 59	▷ dolny, 47
<code>eqnarray</code> , 53	<code>ghostscript</code> , 9, 65, 74	▷ górny, 47
<code>eqnarray*</code> , 53	<code>ghostview</code> , 65	<code>\indent</code> , 108
<code>\eqref</code> , 46	<code>gimp</code> , 66	<code>indentfirst</code> , 17, 108
equation, 46, 53	<code>.glo</code> (plik), 73	<code>\index</code> , 70, 71
<code>eucal</code> , 64	<code>gnuplot</code> , 65	<code>\input</code> , 14, 122
<code>eufrak</code> , 64	<code>graphicx</code> , 11, 65, 67, 76, 81	<code>inputenc</code> , 12, 20, 26–29
<code>eurosym</code> , 23	<code>grave</code> , 25	<code>.ins</code> (plik), 13, 72, 73
<code>\EURtm</code> , 24	<code>grupa</code> , 103	<code>\institute</code> , 83
<i>executivepaper</i> , 10	<code>gsview</code> , 65	<code>\int</code> , 50
<code>exscale</code> , 12, 51	<code>gv</code> , 65	interlinia, 107
<code>\extracolsep</code> , 117	<code>\headheight</code> , 112	▷ podwójna, 107
<code>fancyhdr</code> , 71	<code>\headsep</code> , 112	<code>\item</code> , 36
		<code>itemize</code> , 36, 83

Spis treści

Podziękowania	iii
Przedmowa	v
1. Podstawy, które warto znać	1
1.1. Nazwa programu	1
1.1.1. \TeX	1
1.1.2. \LaTeX	1
1.2. Podstawy	2
1.2.1. Autor, redaktor i zecer	2
1.2.2. Układ graficzny	3
1.2.3. Zalety i wady	3
1.3. Plik źródłowy	4
1.3.1. Odstęp	4
1.3.2. Znaki specjalne	5
1.3.3. Polecenia \LaTeX a	5
1.3.4. Komentarze	6
1.4. Struktura pliku źródłowego	6
1.5. Typowa sesja pracy z \LaTeX em	7
1.6. Układ graficzny dokumentu	9
1.6.1. Klasy dokumentów	9
1.6.2. Pakiety	10
1.6.3. Style strony	11
1.7. Nazwy plików związanych z \LaTeX em	12
1.8. Duże dokumenty	14
2. Składanie tekstu	15
2.1. Struktura tekstu i języka	15
2.2. Składanie akapitów i łamanie stron	17
2.2.1. Składanie akapitów	17
2.2.2. Przenoszenie wyrazów	19
2.3. Kilka gotowych oznaczeń napisów	20
2.4. Znaki specjalne i symbole	21

- [14] L^AT_EX3 Project Team: *L^AT_EX 2_ε for Class and Package writers*. Dokument dostępny w pliku `clsguide.tex` w dystrybucji L^AT_EX 2_ε.
- [15] L^AT_EX3 Project Team: *L^AT_EX 2_ε Font selection*. Dokument dostępny w pliku `fontguide.tex` w dystrybucji L^AT_EX 2_ε.
- [16] *L^AT_EX Local Guide*: Każda wielodostępna instalacja L^AT_EX-owa powinna zawierać *L^AT_EX Local Guide*, w którym są opisane rzeczy specyficzne dla danej lokalnej instalacji. Dokument ten powinien być zawarty w pliku `local.tex`. W wielu wypadkach administratorzy nie udostępniają jednak użytkownikom takiego dokumentu. Pozostaje wtedy zwrócić się o pomoc do lokalnego L^AT_EX-owego guru.
- [17] Lichoński Bogusław: *T_EX na indeksie*. Biuletyn GUST 1994 (3), ISSN: 1230-5650, Dostępny także w `ftp://ftp.gust.org.pl/pub/GUST/bulletin/03/02-bl.pdf`.
- [18] Macewicz Włodzimierz: Wirtualna Akademia T_EXowa. Dostępna w `http://www.ia.pw.edu.pl/~wujek/tex/`.
- [19] Myszka Wojciech: *Włączanie grafik do tekstów w L^AT_EX 2_ε*, Dostępny w `http://www.immt.pwr.wroc.pl/~myszka/grafika/grafika.pdf`.
- [20] Nowacki Janusz M.: *T_EXnologia a typografia*. Biuletyn GUST 1995 (6), ISSN: 1230-5650, Dostępny także w `ftp://ftp.gust.org.pl/pub/GUST/bulletin/06/01-jmn.pdf`.
- [21] Oswald Urs: *Graphics in L^AT_EX 2_ε*, containing some Java source files for generating arbitrary circles and ellipses within the `picture` environment, and *MetaPost – A Tutorial*. Dostępne w `http://www.ursoswald.ch`.
- [22] Reckdahl Keith: *Using EPS Graphics in L^AT_EX 2_ε Documents*. Dostępny w CTAN://`info/epslatex.ps`.
- [23] Rose Kristoffer H.: *X_y-pic User's Guide*. Dostępne w CTAN z pakietem X_y-pic.
- [24] Sapijaszko Grzegorz: *Tworzenie dokumentów pdf przy pomocy L^AT_EXa*, Dostępny w `http://www.sapijaszko.net/pedeefy.pdf`.
- [25] Schöpf Rainer, Raichle Bernd, Rowley Chris: *A New Implementation of L^AT_EX's verbatim Environments*. Dokument dostępny w zestawie pakietów „tools” w pliku `verbatim.dtx`.
- [26] Volovich Vladimir, Lemberg Werner i L^AT_EX3 Project Team: *Cyrillic languages support in L^AT_EX*. Rozpowszechniany w dystrybucji L^AT_EX 2_ε w pliku `cyrguide.tex`.
- [27] Williams Graham: *The T_EX Catalogue* (katalog pakietów dla T_EXa oraz L^AT_EXa). Dokument dostępny w CTAN://`help/Catalogue/catalogue.html`.
- [28] Woliński Marcin: MWCLS *Moje własne klasy dokumentów dla L^AT_EX 2_ε*. *Podręcznik użytkownika*. Dokument rozpowszechniany razem z zestawem klas `mwcls.zip`, por. `http://www.mimuw.edu.pl/~wolinski/mwcls.html`.

4.5. Pakiet <code>verbatim</code>	72
4.6. Instalowanie dodatkowych pakietów	72
4.7. Tworzenie plików PDF: <code>pdfL^AT_EX</code> i <code>hyperref</code>	74
4.7.1. Tworzenie dokumentów PDF w L ^A T _E Xu	74
4.7.2. Fonty bitmapowe i obwiedniowe	75
4.7.3. Dołączanie grafiki	76
4.7.4. Łącza hipertekstowe	77
4.7.5. Problemy z łączami	79
4.7.6. Problemy z zakładkami	79
4.7.7. Interpretacja pliku źródłowego przez L ^A T _E Xa i <code>pdfL^AT_EXa</code>	80
4.7.8. Wymiary kartki papieru	81
4.8. Tworzenie prezentacji	81
4.9. Pakiet <code>pdfscreen</code>	83
5. Tworzenie grafiki matematycznej	84
5.1. Przegląd	84
5.2. Otoczenie <code>picture</code>	85
5.2.1. Podstawowe polecenia	85
5.2.2. Odcinki	86
5.2.3. Strzałki	87
5.2.4. Okręgi	88
5.2.5. Tekst i wzory	89
5.2.6. Polecenia <code>\multiput</code> i <code>\linethickness</code>	89
5.2.7. Owale	90
5.2.8. Wielokrotne użycie pudełek z rysunkami	91
5.2.9. Krzywe Béziera drugiego stopnia	92
5.2.10. Krzywe łańcuchowe	93
5.2.11. Prędkość w Szczególnej Teorii Względności	94
5.3. X _y -pic	94
6. Adaptowanie L^AT_EXa	98
6.1. Definiowane instrukcji i otoczeń	98
6.1.1. Instrukcje definiowane przez użytkownika	99
6.1.2. Otoczenia definiowane przez użytkownika	100
6.1.3. Nadmiarowe odstęp	101
6.1.4. Własne pakiety	101
6.2. Fonty	102
6.2.1. Instrukcje przełączające stopień pisma	102
6.2.2. Uwaga, niebezpieczeństwo!	105
6.2.3. Użycie alternatywnych krojów pisma	105
6.3. Odstępy	107
6.3.1. Zmiana wielkości interlinii	107
6.3.2. Odstępy wokół akapitów	108
6.3.3. Odstępy poziome	108

6.8.4. Pakiet tap

Pakiet tap (autorzy: Bogusław Jackowski, Piotr Pianowski i Piotr Strzelczyk) pozwala na eleganckie składanie nawet najbardziej ekstrawaganckich (jednostronicowych) tabel, z tym że stosuje niestandardowy zapis¹¹, co może powodować problemy w integracji z innymi pakietami \LaTeX a, np. pakietem colortbl, służącym do kolorowania rubryk w tabelach. Oto przykład:

```
\input{tap.tex} %% <- dołączenie pakietu
\desiredwidth=\textwidth %% skład na szerokość łamu
\begin{table}
\begin{tableformat \left " &\right \end{tableformat}
\=
\B!^ | @2\center{Wys. w~m~n.p.m.} | Długość |
      @2\center{Nachylenie \%} \E!
\B"- \center{Nazwa} " @2\-" " @2\-" \E"
\B"_ | początek | koniec | w~km | śr. | max \E!
\B! Col du Galibier | -- | 2645 | 17,5 | 6,9\% | 14,5 \E!
\B! Alpe D'Huez | -- | 1839 | 13,0 | 8,5\% | 15,0 \E!
\B! Passo Gavia | -- | 2621 | 17,3 | 7,9\% | 20,0 \E!
\=
\end{table}
```

Wynik jest imponujący:

Nazwa	Wys. w m n.p.m.		Długość w km	Nachylenie %	
	początek	koniec		śr.	max
Col du Galibier	1401	2646	18,1	6,9%	14,5
Alpe D'Huez	724	1815	14,2	7,7%	15,0
Passo Gavia	1734	2618	13,5	6,5%	20,0

Więcej informacji można odnaleźć w (polskojęzycznej) dokumentacji pakietu (CTAN://macros/generic/tables/tap077.zip).

Na zakończenie uwaga: składanie tabel w \LaTeX u nie jest może aż tak proste jak w edytorach WYSIWYG, ale za to można tworzyć konstrukcje niezwykle trudne lub wręcz niemożliwe do wykonania w innych programach. Ponadto zadanie składania tabel można sobie znakomicie ułatwić, stosując skryptowe języki programowania, w rodzaju Perla czy Pythona. Już stosunkowo niewielka umiejętność programowania w wymienionych językach pozwala na szybką konwersję generowanych przez zewnętrzne aplikacje danych tabelarycznych do formatu \LaTeX a.

¹¹Tak naprawdę, nie jest to nawet pakiet w rozumieniu \LaTeX a i zamiast poleceniem `\usepackage` należy go dołączać poleceniem `\input`.

Spis rysunków

1.1. Zawartość minimalnego pliku źródłowego	7
1.2. Przykład artykułu do czasopisma	8
4.1. Przykład wykorzystania pakietu fancyhdr	71
4.2. Prosty kod dla klasy beamer	82
6.1. Przykładowy pakiet	102
6.2. Parametry układu graficznego strony	112

To jest nagłówek następnych stron			
k-1	k-2	k-3	k-4
Alpe D'Huez	1815	14,2	7,7%
Col de la Croix de Fer	2067	22,0	7,0%
Col de Portet d'Aspet	1069	4,4	9,6%
Mont Ventoux	1912	21,0	7,6%
Col de l'Izoard	2361	15,9	6,9%
Passo dello Stelvio	2758	24,3	7,4%
Col du Trébuchet	1143	14,6	4,3%
Passo Gavia	2618	13,5	6,5%
Col du Grand Colombier	1505	15,9	7,8%
Col de l'Iseran	2770	48,0	4,1%
Courchevel	2000	21,7	6,5%
Col de l'Aspin	1489	12,0	6,5%
Col de la Croix de l'Homme Mort	1163	18,0	4,3%
Col de Notre Dame des Abeilles	1000	13,9	5,1%
Paso del Morredero	1872	25,0	5,4%
Collado de la Caballar	1308	14,5	7,5%
Capilla de la Magdalena	1137	11,0	5,5%
La Bola del Mundo	2257	21,8	6,2%
Stopka na ostatniej stronie			

6.8.3. Pakiet array

Do składu tabel o bardziej skomplikowanym układzie graficznym lepiej używać pakietu `array`. Pakiet ten nie wprowadza żadnego nowego otoczenia, redefiniuje jedynie i rozszerza standardowe otoczenie `tabular`.

W „standardowym” otoczeniu `tabular` konstrukcja `p{szer-kolumn}` deklaruje kolumnę, w której zawartość każdej rubryki będzie składana w prostokąt o zadanej szerokości, z wyrównywaniem obu marginesów (odpowiednik `\parbox[t]{szer}`). Pakiet `array` wprowadza ponadto `b{szer}` – odpowiednik `\parbox[b]{szer}` – oraz `m{szer}` – odpowiednik `\parbox{szer}`. Przykład:

```
\begin{tabular}{|p{8mm}|p{8mm}|p{8mm}|} \hline
x y z x y z x y z x y z & x y z x y z x y y&
1 1 1 1 1 \\\ \hline \end{tabular}
\begin{tabular}{|m{8mm}|m{8mm}|m{8mm}|} \hline
x y z x y z x y z x y z & x y z x y z x y y&
2 2 2 2 2 \\\ \hline \end{tabular}
\begin{tabular}{|b{8mm}|b{8mm}|b{8mm}|} \hline
x y z x y z x y z x y z & x y z x y z x y y&
3 3 3 3 3 \\\ \hline
\end{tabular}
```

Rozdział 1

Podstawy, które warto znać

W pierwszej części tego rozdziału przedstawimy krótko filozofię oraz historię systemu \LaTeX 2 ϵ . W części drugiej skoncentrujemy się na podstawowych elementach dokumentu \LaTeX owego. Po przeczytaniu tego rozdziału czytelnik powinien z grubsza wiedzieć, jak działa \LaTeX , co jest niezbędne do rozumienia materiału prezentowanego w następnych rozdziałach.

1.1. Nazwa programu

1.1.1. \TeX

\TeX jest programem komputerowym stworzonym przez Donalda E. Knutha [11]. Jest przeznaczony do składu tekstów oraz wzorów matematycznych. Knuth rozpoczął pracę nad \TeX em w 1977 roku, aby wykorzystać potencjał składu cyfrowego, stosowanego wówczas na coraz szerszą skalę w poligrafii. Miał też nadzieję, że uda się odwrócić tendencję do pogarszania się jakości typograficznej, co uwidaczniało się w jego własnych książkach i artykułach. W używanej obecnie postaci \TeX został udostępniony w roku 1982, a niewielkie rozszerzenie, dotyczące ośmiobitowego kodowania znaków, pojawiło się w roku 1989. \TeX ma renomę programu nadzwyczaj stabilnego, pracującego na różnego rodzaju sprzęcie oraz praktycznie wolnego od błędów. Numery wersji \TeX a zbiegają do liczby π , a obecny wynosi 3,14159.

Słowo \TeX należy wymawiać „tech”. Zgłoska „ch” ma związek z tym, że znak X przypomina grecką literę „chi”. \TeX jest też pierwszą sylabą greckiego słowa technologia (technologia). W sytuacjach, w których nie można zapisywać nazwy \TeX z charakterystycznym obniżeniem litery E, należy zamiennie używać wersji `TeX`.

1.1.2. \LaTeX

\LaTeX jest zestawem instrukcji (poleceń, makrodefinicji, makr) umożliwiających autorom złożenie i wydrukowanie ich prac na najwyższym poziomie

kolumnami 2–3 oraz 5–6 nie są „dociągnięte” do lewego brzegu. Niestety nie da się tego naprawić – przynajmniej nie w prosty sposób. Otoczenie `tabular*` nadaje się do składania tabel, ale tylko wtedy, gdy nie korzystamy zbyt często z polecenia `\cline`.

Spróbujmy teraz zastosować otoczenie `tabularx`. W tym celu najpierw do preambuły dokumentu dołączamy pakiet `tabularx`. Następnie zmieniamy tabelę w taki sposób (kolumny, których szerokość ma być wyznaczona automatycznie, oznaczamy symbolem `X`):

```
\begin{tabularx}{\textwidth}{|X|X|X|X|X|}\hline
...
\end{tabularx}
```

Złożona tabela wygląda następująco:

Nazwa	Wys. w m n.p.m.		Długość w km	Nachylenie %	
	początek	koniec		śr.	max
Col du Galibier	1401	2646	18,1	6,9%	14,5
Alpe D'Huez	724	1815	14,2	7,7%	15,0
Passo Gavia	1734	2618	13,5	6,5%	20,0

Teraz wprowadź wszystkie kreski są elegancko dociągnięte, ale jest problem z pierwszą kolumną: jest ona zbyt wąska, a przez to zawartość rubryk już się nie mieści i musi zostać przeniesiona⁹. Otoczenie `tabularx` automatycznie dzieli bowiem tabelę wyłącznie na kolumny *o równej szerokości*. Jest to jego największe ograniczenie funkcjonalne.

Specyfikacja kolumn tabeli oprócz `X` może zawierać także wartości „tradycyjne”, takie jak `l`, `r` lub `c`. Odpowiadającym tym specyfikacjom kolumny mają naturalną szerokość. Przykładowo:

```
\begin{tabularx}{\textwidth}{|l|X|X|X|X|}\hline
...
\end{tabularx}
```

Akurat w naszym przykładzie osiągnęliśmy zadowalający rezultat, gotowa tabela wygląda bowiem następująco:

Nazwa	Wys. w m n.p.m.		Długość w km	Nachylenie %	
	początek	koniec		śr.	max
Col du Galibier	1401	2646	18,1	6,9%	14,5
Alpe D'Huez	724	1815	14,2	7,7%	15,0
Passo Gavia	1734	2618	13,5	6,5%	20,0

⁹Ponadto lepiej, by kolumny 2–6 były wyrównane do prawego, a nie do lewego brzegu rubryki.

dokument na ekranie po przetworzeniu go L^AT_EXem². Gdy jest już gotowy, można dokument wysłać na drukarkę.

1.2.2. Układ graficzny

Projektowanie książek jest sztuką. Amatorzy często popełniają poważny błąd, zakładając, że zaprojektowanie układu graficznego książki jest jedynie kwestią estetyki (jeżeli dokument ładnie wygląda, to jest dobrze złożony). Ponieważ dokumenty są przeznaczone do czytania, a nie do powieszenia, jak obraz na ścianie w galerii, to o wiele większe znaczenie niż piękny wygląd ma łatwość czytania i przyswajania tekstu. Przykłady:

- stopień pisma oraz numerację rozdziałów, podrozdziałów i punktów należy ustalić tak, by czytelnik mógł się szybko zorientować w strukturze dokumentu;
- szerokość szpalty powinna być na tyle wąska, by czytelnik nie musiał wyteżać wzroku, wystarczająco jednak duża, aby tekst elegancko wypełniał stronę.

W systemach wizualnych często powstają dokumenty przyjemne dla oka, chociaż pozbawione struktury albo wykazujące brak konsekwencji w strukturze. L^AT_EX zapobiega powstawaniu takich błędów, nakłaniając autora do określenia *logicznej* struktury dokumentu. Do L^AT_EXa należy dobór najbardziej odpowiedniego dla niej układu graficznego.

1.2.3. Zalety i wady

Tematem często dyskutowanym, gdy użytkownicy programów typu WYSIWYG spotykają użytkowników L^AT_EXa, są „zalety L^AT_EXa w porównaniu ze zwykłym procesorem tekstu” albo na odwrót. Najlepiej podczas takich dyskusji siedzieć cicho. Czasami jednak nie ma ucieczki. . .

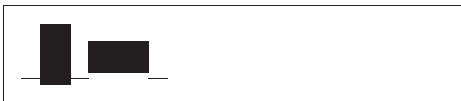
Na wszelki wypadek trochę amunicji. Oto najważniejsze zalety L^AT_EXa w porównaniu ze zwykłymi procesorami tekstu:

- Dostępne są gotowe, przygotowane przez zawodowców szablony, dzięki zastosowaniu których dokumenty wyglądają „jak z drukarni”.
- Wygodnie składa się wzory matematyczne.
- Do rozpoczęcia pracy wystarczy poznać zaledwie kilkanaście łatwych do zrozumienia instrukcji, określających strukturę logiczną dokumentu. Nie trzeba zaprzętać sobie głowy formatowaniem dokumentu.
- Nawet takie elementy jak: przypisy, odnośniki, spisy treści, spisy tabel, skorowidze oraz spisy bibliograficzne przygotowuje się bardzo łatwo.

²Na szybkim komputerze przetworzenie trwa często zaledwie kilka sekund. Dysponując dużym ekranem, można jednocześnie wyświetlić okno z plikiem źródłowym oraz okno podglądu, otrzymując w ten sposób system prawie WYSIWYG.

otrzymujemy w składzie czarny prostokąt:

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



Polecenie `\rule` służy zwykle do rysowania kresek pionowych i poziomych. Na przykład gruba czarna krecha na stronie tytułowej niniejszego *Wprowadzenia* to wynik zadziałania instrukcji:

```
\rule[-1ex]{\textwidth}{5pt}
```

Parametr *przesunięcie* określa, jak wysoko przesunąć kreskę ponad linię podstawową (lub opuścić poniżej linii podstawowej, jeśli parametr jest ujemny).

Specjalnym przypadkiem jest kreska o zerowej szerokości, lecz o niezerowej wysokości. Taką kreskę nazywamy *podporą* (*strut*). Podpora to często stosowana metoda nadawania wszystkim elementom jednakowej wysokości. Spójrzmy na poniższy przykład. Dzięki wstawieniu podpory drugi wiersz ma tę samą wysokość co pierwszy. Zawartość wiersza trzeciego jest identyczna jak drugiego, ale w drugim występuje niewidoczna podpora. Gdyby jej brakowało, to wysokość drugiego wiersza byłaby mniejsza.

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

Pittprop ...
Strut
Strut

6.8. Więcej o składaniu tabel

6.8.1. Tabele o zadanej szerokości

W otoczeniu `tabular` szerokość poszczególnych kolumn tabeli jest ustalana automatycznie, a szerokość tabeli jest sumą szerokości kolumn i odstępów międzykolumnowych. Czasami jednak trzeba złożyć tabelę o z góry zadanej szerokości; jest to potrzebne przykładowo wtedy, gdy wszystkie tabele w dokumencie mają mieć tę samą szerokość.

Do tego celu można użyć albo otoczenia `tabular*`, albo otoczenia `tabularx` z pakietu o tej samej nazwie. Niestety każdy z tych sposobów ma poważne ograniczenia funkcjonalne. Rozpocznijmy od otoczenia `tabular*`, które różni się od `tabular` tylko jednym dodatkowym parametrem obowiązkowym, określającym szerokość tabeli:

```
\begin{tabular*}{szerokość}{spec-kolumn}
```

Nie ma znaczenia, czy między słowami jest jedna czy więcej spacji.

Nie ma znaczenia, czy między słowami jest jedna czy więcej spacji.
Pusty wiersz zakończył poprzedni akapit.

Pusty wiersz zakończył poprzedni akapit.

1.3.2. Znaki specjalne

Niektóre znaki są zarezerwowane – w tym sensie, że albo mają dla L^AT_EXa specjalne znaczenie, albo nie są dostępne we wszystkich standardowych krojach pisma. Użyte dosłownie w pliku źródłowym nie pojawią się na wydruku, lecz (najczęściej) spowodują błąd podczas przetwarzania tekstu. Oto ich lista:

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

Znaki te można umieścić w dokumencie pod warunkiem, że w pliku źródłowym zostaną poprzedzone znakiem w-tył-ciacha (ang. *backslash*):

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

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

Samego znaku w-tył-ciach *nie można* wstawić do tekstu metodą podwojenia, kombinacja `\\` jest bowiem poleceniem L^AT_EXa, opisanym w punkcie 2.2.1. Znak w-tył-ciach można wstawić poleceniem `\\backslash$`. Uwaga: znaki dolara są tu niezbędne, a ich opuszczenie spowoduje błąd podczas przetwarzania.

1.3.3. Polecenia L^AT_EXa

Polecenia L^AT_EXa mogą wystąpić w dwóch następujących odmianach:

1. Instrukcji rozpoczynających się znakiem w-tył-ciach „`\`”, po którym występuje ciąg liter. Końcem instrukcji jest wówczas odstęp lub inny znak niebędący literą. W nazwach instrukcji L^AT_EX rozróżnia litery małe i duże, nie można też w nich używać polskich liter diakrytycznych.
2. Instrukcji składających się ze znaku w-tył-ciach oraz *jednego* znaku niebędącego literą.

L^AT_EX ignoruje znaki niewidoczne po instrukcji typu 1. Jeżeli po instrukcji ma występować w dokumencie odstęp, to należy bezpośrednio po niej umieścić kolejno: parę nawiasów klamrowych `{}` i odstęp. Para znaków `{}` zapobiega zignorowaniu przez L^AT_EXa odstępów po nazwie instrukcji. Innym sposobem jest wstawienie specjalnej instrukcji `\` (tj. w-tył-ciach i spacja). Niektórzy zapobiegają „połykaniu” spacji w jeszcze inny sposób, a mianowicie otaczając nazwę instrukcji parą nawiasów `{ }`.

łatwo zestawić obok siebie. Trzeba jedynie zadbać o to, by łączna szerokość połączonych obiektów nie przekraczała szerokości szpalty.

To samo odnosi się do akapitów, które – jeśli tego potrzebujemy – możemy składać w pudełka o zadanej szerokości:

```
\parbox[pos]{szerokość}{tekst}
```

Do tego samego celu można wykorzystać otoczenie:

```
\begin{minipage}[pos]{szerokość} tekst \end{minipage}
```

Argument *szerokość* to wymiar określający szerokość pudełka. Argument *pos* jest jednoliterowy i może przyjmować jedną z wartości: c, t lub b. Wartości te określają, jak \LaTeX ma umieścić pudełko względem otaczającego tekstu. Wartość c oznacza umieszczenie środka wysokości pudełka na linii podstawowej, t – umieszczenie linii podstawowej pierwszego wiersza w pudełku na linii podstawowej otaczającego tekstu, natomiast b – umieszczenie dolnej krawędzi pudełka na linii podstawowej. Wynik zastosowania parametrów ilustruje poniższy przykład (linię podstawową oznaczono kreską):

```
\makebox[0pt][l]{\rule{66mm}{.4pt}}%
\parbox[c]{9mm}{5 5 5 5 5 5 5}
\parbox[t]{9mm}{6 6 6 6 6 6 6}
\parbox[b]{9mm}{8 8 8 8 8 8 8}
```

```

      8 8 8
5 5 5   8 8 8
5 5 5 6 6 6 8 8 8
5 5 5 6 6 6
      6 6 6
```

Polecenie `\parbox` składa tekst w pudełku, w razie potrzeby dzieląc tekst na linijki. Inaczej jest z pudełkami uzyskiwanymi poleceniem `\mbox`, których zawartość nigdy nie jest dzielona na wiersze. Polecenia tego używamy, gdy trzeba zapobiec dzieleniu wyrazu lub sekwencji wyrazów na wiersze. Polecenie `\mbox` jest uproszczoną wersją instrukcji `\makebox`, o składni:

```
\makebox[szerokość][pos]{tekst}
```

W opcjonalnym argumencie *szerokość* możemy zadać szerokość pudełka. Może się ona różnić od naturalnej szerokości tekstu w pudełku; może wynosić zero, a nawet być wielkością ujemną! W obrębie argumentu *szerokość* możemy się też posługiwać wielkościami `\width` (szerokość), `\height` (wysokość), `\depth` (głębokość) oraz `\totalheight` (suma wysokości i głębokości). Ponadto argument *pos* określa sposób umieszczenia tekstu. Litera c oznacza wyśrodkowanie, l – dosunięcie do lewej, r – dosunięcie do prawej, a s – wypsacjowanie zawartości.

Poniższy przykład ilustruje użycie polecenia `\width` w obrębie argumentu *szerokość*. Pierwsze pudełko ma szerokość równą połowie naturalnej szerokości tekstu:

Część główna dokumentu zaczyna się od instrukcji `\begin{document}`. Za nią znajduje się tekst dokumentu, wzbogacony o \LaTeX owe polecenia sterujące wyglądem. Na końcu dokumentu musi występować polecenie `\end{document}`. Tekst znajdujący się za tym poleceniem jest przez \LaTeX a ignorowany.

Rysunek 1.1 pokazuje zawartość minimalnego dokumentu \LaTeX owego. Użyte w nim instrukcje `\usepackage{...}`, niezbędne do składania w języku polskim, omawiamy w punkcie 2.5.

```

\documentclass{article}
\usepackage[MeX]{polski}
% kodowanie: latin2, utf8 lub cp1250
\usepackage[latin2]{inputenc}
\begin{document}
Maże jest piękne.
\end{document}
```

Rysunek 1.1: Zawartość minimalnego pliku źródłowego

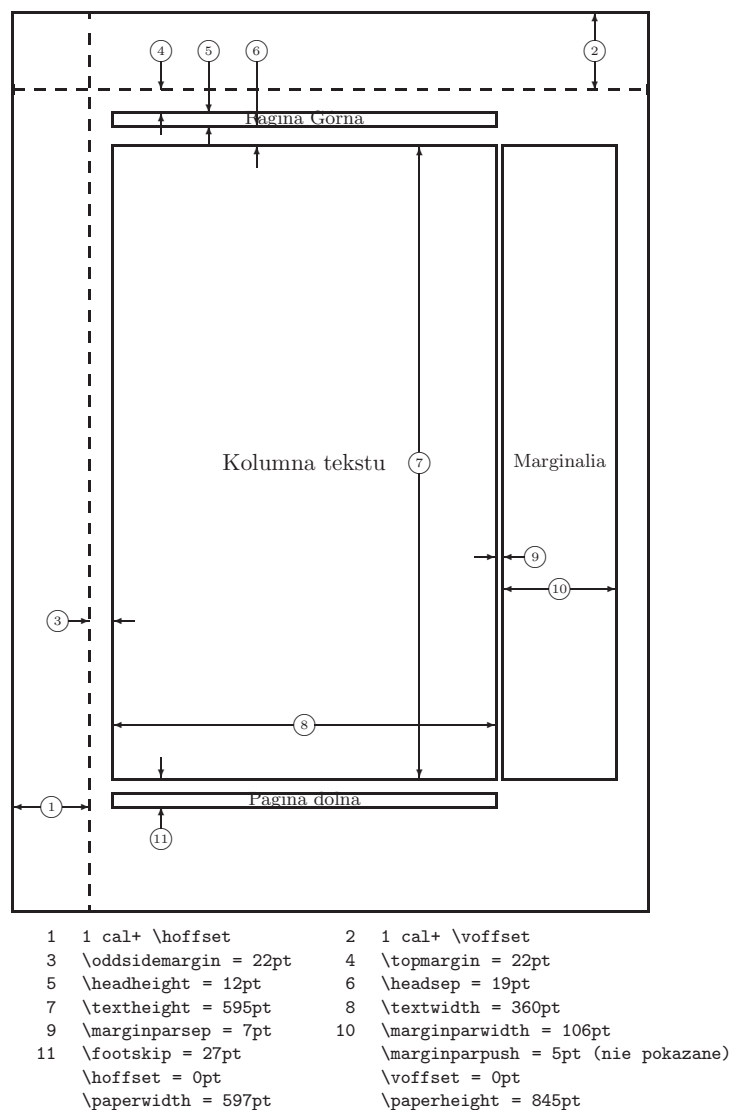
Rysunek 1.2 przedstawia nieco bardziej rozbudowany plik źródłowy.

1.5. Typowa sesja pracy z \LaTeX em

Na pewno chciałbyś już sprawdzić, jak będzie wyglądał na papierze dokument z przykładu 1.1. Szczegółowy sposób uruchomienia \LaTeX a zależy od systemu operacyjnego, wersji i upodobań użytkownika⁴. \LaTeX „jako taki” nie jest wyposażony w zintegrowane środowisko graficzne (IDE). Praca z takim systemem polega na wydawaniu odpowiednich poleceń z wiersza poleceń systemu operacyjnego. Oczywiście posługiwanie się \LaTeX em w ten sposób jest na dłuższą metę niewygodne, niemniej do stworzenia pierwszego dokumentu nic więcej nie potrzeba. Dzięki temu będziesz też wiedział, co kryje się za guzikami, gdy później zainstalujesz jedno z istniejących zintegrowanych środowisk graficznych do pracy z \LaTeX em.

1. Uruchom edytor i wpisz tekst z przykładu 1.1. Zapisz dokument jako tekstowy plik ASCII. W systemach typu Unix możesz do tego użyć praktycznie dowolnego edytora. W systemie MS Windows musisz zapisać plik jako *Zwykły tekst*, *Dokument tekstowy* lub wybrać *Wszystkie pliki* jako typ dokumentu. W różnych systemach obowiązują różne warianty kodowania polskich znaków. Najpopularniejsze z tych wariantów, to: kodowanie wielobajtowe *Unicode* (MS Windows,

⁴O tym, jak zainstalować \LaTeX a, można przeczytać na <http://www.gust.org.pl>.



Rysunek 6.2: Parametry układu graficznego strony

W wypadku sukcesu zakończy się to utworzeniem pliku o rozszerzeniu `.dvi`⁸. Polecenie trzeba powtórzyć kilka razy, by L^AT_EX mógł utworzyć spis treści i/lub odsyłacze do pozycji bibliograficznych, rysunków, wzorów matematycznych i śródtytułów. Napotkawszy błąd w dokumencie, L^AT_EX zakończy jego przetwarzanie i przejdzie do trybu dialogu z użytkownikiem. Wpisanie `Ctrl-D` (lub `Ctrl-Z` w MS Windows) w tym trybie powoduje powrót na poziom wiersza poleceń.

3. Teraz możesz obejrzeć plik DVI. Jest na to kilka sposobów. W systemie typu Unix wyposażonym w środowisko graficzne X Windows możesz wywołać:

```
xdvi foo.dvi &
```

W systemie MS Windows możesz skorzystać z programu yap bądź innej przeglądarki plików DVI. Możesz też przetworzyć plik DVI do formatu Postscript:

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

Pliki w formacie Postscript można oglądać i drukować za pomocą programu ghostscript.

Jeśli twoja dystrybucja L^AT_EXa zawiera program dvi2pdf do konwersji plików DVI na format PDF, to spróbuj go wywołać, wpisując:

```
dvi2pdf foo.dvi
```

Pliki PDF można oglądać i drukować, korzystając z programu Acrobat lub wspomnianego już ghostscripta.

1.6. Układ graficzny dokumentu

1.6.1. Klasy dokumentów

Na samym początku przetwarzania pliku źródłowego L^AT_EX musi się dowiedzieć, jakiego typu dokument autor chce uzyskać. Określone jest to w instrukcji `\documentclass`:

```
\documentclass[opcje]{klasa}
```

gdzie *klasa* oznacza typ dokumentu. W niniejszym wprowadzeniu opisano następujące klasy ze standardowej dystrybucji L^AT_EX 2_ε⁹:

`article` artykuły, krótkie opracowania. . .

⁸W większości współczesnych dystrybucji systemu T_EX dostępne jest polecenie `pdflatex` do generowania dokumentu w formacie PDF. Więcej na ten temat w punkcie 4.7.

⁹W dystrybucji L^AT_EX 2_ε znajdują się także inne, rzadziej wykorzystywane klasy, np. `slides` do przygotowywania przeźroczy. Zamiast klasy standardowej można do tworzenia slajdów zastosować pakiet `foiltex`, dostępny pod adresem `CTAN://macros/latex/packages/supported/foiltex`, bądź pakiet `beamer`, spod adresu `CTAN://macros/latex/contrib/beamer` (krótkie wprowadzenie do pakietu `beamer` zawiera punkt 4.8).

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



Wielkość odstępów towarzyszących tekstowi warto dostosować do aktualnego rozmiaru czcionki. Można do tego użyć względnych jednostek miary em oraz ex:

```
{\Large{}big\hspace{1em}y} \quad
{\tiny{}tin\hspace{1em}y}
```



Przypominamy, że polecenia `\quad` i `\qquad` wstawiają odstęp poziomy o szerokości, odpowiednio, 1 em oraz 2 em.

6.3.4. Odstęp pionowe

Odstępy pionowe między akapitami, rozdziałami, punktami itp. \LaTeX wstawia automatycznie. Jeśli zachodzi potrzeba wstawienia dodatkowego odstępu pionowego, to należy zastosować polecenie:

```
\vspace{odległość}
```

Polecenie to należy oddzielić pustymi liniami od otaczającego je tekstu. Jeżeli w wyniku złamania strony odstęp taki znajdzie się na początku lub na końcu strony (będzie zaczynał lub też kończył kolumnę tekstu), to zostanie on usunięty. Jeżeli ma zostać wstawiony także na początku lub końcu strony, to należy użyć wersji „gwiazdkowej” `\vspace*`⁶. Argument *odległość* oznacza \LaTeX owy wymiar.

Do rozmieszczania tekstu kolumny w pionie można używać polecenia `\stretch`, łącznie z `\pagebreak`. W poniższym przykładzie tekst zostanie rozmieszczony tak, że odstęp u dołu będzie dwa razy mniejszy od odstępu u góry strony:

```
\vspace{\stretch{1}}
Tytuł i~autor
\vspace{\stretch{2}}\pagebreak
```

Dodatkowy odstęp między dwoma wierszami tego samego akapitu lub między wierszami tabeli możemy uzyskać poleceniem:

```
\[odległość]
```

Polecenia `\bigskip`, `\medskip` i `\smallskip` wstawiają odpowiednio odstępy „elastyczne” o następujących wielkościach: $12\text{pt} \pm 4\text{pt}$, $6\text{pt} \pm 2\text{pt}$ oraz $3\text{pt} \pm 1\text{pt}$ ⁷.

⁶Zwróćmy uwagę, że w takim wypadku wysokość kolumny tekstu na sąsiednich stronach nie będzie jednakowa, stosujemy zatem polecenie `\vspace*` z pewną ostrożnością.

⁷Nie są to wielkości przypadkowe: 12 punktów to typowa odległość między liniami podstawowymi wierszy przy składzie pismem w stopniu 10 punktów.

itp. Do wykonywania tych zadań służą właśnie pakiety. Dołącza się je poleceniem:

```
\usepackage[opcje]{pakiet}
```

gdzie *pakiet* oznacza nazwę pakietu, a *opcje* – listę rozdzielonych przecinkami opcji. Część pakietów znajduje się w podstawowej dystrybucji \LaTeX 2 ϵ (zobacz tabela 1.1), kolejnych kilkanaście, m.in. tak użyteczne jak `babel` czy `graphicx`, jest do niej *obowiązkowo* dodawanych; pozostałe są rozpoznawane oddzielnie. Gdy używamy \LaTeX a w systemie, którym zarządza (dobry) administrator, to informacja o dostępnych pakietach powinna się znajdować w [16]. Podstawowym źródłem informacji o \LaTeX u jest [6]. Zawiera on opis setek pakietów, a także informuje, jak można pisać własne rozszerzenia \LaTeX a.

Wartościowym źródłem informacji o istniejących pakietach \LaTeX a jest *TeX Catalogue Online* Grahama Williama [27], dostępny niestety tylko w języku angielskim. Użytkownikom polskojęzycznym polecamy *Wirtualną Akademię* Włodzimierza Macewicz [18].

1.6.3. Style strony

Typowa strona składa się z trzech podstawowych części. Powyżej kolumny tekstu głównego znajduje się *pagina górna* (główka), która może zawierać numer strony, tytuł rozdziału czy punktu. Poniżej kolumny tekstu znajduje się *pagina dolna* (stopka). W niniejszym wprowadzeniu pagina dolna jest pusta, górna natomiast zawiera numer strony oraz tytuł rozdziału na stronach parzystych, a tytuł punktu – na nieparzystych¹⁰.

\LaTeX pozwala wybrać jeden z trzech sposobów składania pagin. Służy do tego instrukcja:

```
\pagestyle{styl}
```

Dopuszczalne wartości argumentu *styl* są następujące:

plain pagina górna jest pusta, a pagina dolna zawiera wycentrowany numer strony. Ten styl jest domyślny;

headings pagina górna zawiera numer strony oraz tytuł, pagina dolna jest pusta;

empty pagina górna i dolna są puste.

Możliwa jest także zmiana stylu *bieżącej* strony. Służy do tego polecenie:

```
\thispagestyle{styl}
```

¹⁰Paginę zawierającą oprócz kolejnego numeru kolumny (strony) także informację dotyczącą treści tej kolumny drukarze nazywają *żywą paginą*.

6.3.2. Odstępów wokół akapitów

Dwa dodatkowe parametry określają w L^AT_EXu wielkość, odpowiednio, wcięcia akapitowego oraz odstęp między akapitami. Wpisując na przykład do preambuły dokumentu:

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

ustalamy wielkość wcięcia akapitowego na 0 pt (co powoduje, że akapity będą się zaczynać bez wcięć), a odstęp między akapitami ustalamy na 1 ex plus 0,5 ex minus 0,2 ex. Drugi zapis oznacza, że normalny odstęp między akapitami, wynoszący 1,0 ex (jednostki miary w L^AT_EXu podaje tabela 6.5 na stronie 109), może się zwiększyć do 1,0 + 0,5 = 1,5 ex lub zmniejszyć do 1,9 – 0,2 = 0,8 ex. W Europie kontynentalnej akapity składa się czasami bez wcięcia akapitowego, a jedynie z dodatkowym odstępem między nimi. Ale uwaga! Ten efekt pojawi się także w spisie treści, tabel i rysunków, gdzie poszczególne pozycje spisu będą od siebie bardziej oddalone (w spisach większość akapitów jest jednowierszowa). Aby uniknąć tego trochę śmiesznego efektu, należy usunąć powyższe instrukcje `\setlength` z preambuły dokumentu i wstawić je w części głównej, po poleceniach `\tableofcontents` itp. Najlepiej jednak wcale nie korzystać z tego sposobu, gdyż znakomita większość książek jest składana z wcięciem akapitowym, a nie z dodatkowymi odstępami między akapitami⁴.

Wcięcie akapitowe na początku akapitu wstawiamy poleceniem⁵:

```
\indent
```

Wstawienie `\indent`, kiedy wartość `\parindent` wynosi zero, nie przyniesie oczywiście żadnego efektu.

Aby uzyskać akapit bez wcięcia, należy przed nim umieścić polecenie:

```
\noindent
```

6.3.3. Odstępów poziome

Wielkość odstępów między słowami oraz między zdaniem L^AT_EX ustala automatycznie. Dodatkowy odstęp poziomy (przez *odstęp poziomy* rozumiemy odstęp między wyrazami, przez *odstęp pionowy* – odstęp między wierszami i akapitami) możemy wstawić poleceniem:

```
\hspace{odległość}
```

⁴Jednoczesne użycie wcięć i powiększonych odstępów między akapitami uważa się w Polsce za poważny błąd typograficzny.

⁵Dla uzyskania efektu wcięcia w pierwszym akapicie po tytule rozdziału, punktu itd. należy dołączyć do dokumentu pakiet `indentfirst` z zestawu pakietów „tools”.

- `.dtx` Udokumentowany T_EX. Jest to podstawowy format, w jakim dystrybuowane są style L^AT_EXa. Skutkiem kompilacji pliku tego typu jest broszurka z udokumentowanymi makrami.
- `.ins` Instalator dla plików `.dtx`. Ściągając z sieci pakiet L^AT_EXowy, otrzymasz na ogół pliki `.dtx` i `.ins`. Uruchomienie L^AT_EXa na pliku `.ins` powoduje rozpakowanie pliku `.dtx`.
- `.cls` Plik z klasą L^AT_EXa definiującą wygląd składanych w L^AT_EXu dokumentów. Właśnie do tych plików odnosi się występująca na początku dokumentu instrukcja `\documentclass`.
- `.fd` Definicja niektórych właściwości fontów L^AT_EXa.
 - W wyniku kompilacji dokumentu powstają następujące pliki:
- `.dvi` *Device Independent File* (plik niezależny od urządzenia), będący wynikiem kompilacji pliku źródłowego przez „tradycyjnego” L^AT_EXa¹¹. Zawartość plików DVI możesz zobaczyć w przeglądarce plików DVI albo posłać na drukarkę, korzystając z programu `dvips` albo innego programu o podobnych funkcjach.
- `.pdf` Portable Document Format (przenośny format dokumentów). Zagadnienie generowania plików w formacie PDF jest omawiane w punkcie 4.7, zaczynającym się na stronie 74.
- `.log` Zawiera szczegółowy raport z tego, co się wydarzyło podczas kompilacji: które pliki były przetwarzane, co szczególnego i ewentualnie jakie błędy L^AT_EX w nich znalazł, a także – jakie pliki powstały w wyniku kompilacji.
- `.toc` Zawiera nagłówki rozdziałów i punktów dokumentu. Jest czytany przez L^AT_EXa w następnym przebiegu kompilacji, w celu wygenerowania spisu treści.
- `.lof` Podobny do pliku `.toc`, z tym że zawiera wykaz ilustracji.
- `.lot` Tak samo, lecz dotyczy wykazu tabel.
- `.aux` Inny plik pomocniczy, przenoszący informację z jednego przebiegu kompilacji do następnego. Jest używany między innymi do magazynowania informacji związanej z odsyłaczami występującymi w dokumencie.
- `.idx` Jeśli dokument zawiera skorowidz, to w tym pliku L^AT_EX zapisze wszystkie jego hasła. Do przetworzenia tego pliku służy program `makeindex` (lub `plmindex`, w przypadku języka polskiego). Więcej o tworzeniu skorowidzów przeczytasz w punkcie 4.3 na stronie 69.
- `.ind` Przetworzony plik `.idx`, gotowy do włączenia do dokumentu w następnym cyklu kompilacji.
- `.ilg` Sprawozdanie z tego, co zrobił program `makeindex`.

¹¹W nowych wersjach systemu, L^AT_EX potrafi także generować dokumenty bezpośrednio w formacie PDF. Więcej informacji na ten temat zawiera punkt 4.7.

rodzina *Computer Modern*, często na przykład brakuje w nich kompletu symboli matematycznych, znaków z alfabetów nielacińskich, takich jak greka lub alfabet cyrylicy, albo niektórych odmian, jak na przykład kapitalików.

Z drugiej strony krój CM ma też wady: kreski znaków są cieńsze, a względna wysokość małych liter² jest mniejsza niż w wielu innych krojach. Te cechy kroju CM powodują, że jest mniej czytelny w wypadku, gdy jest reprodukowany na nośniku o niskiej rozdzielczości, lub – mówiąc wprost – nie najlepiej się nadaje do dokumentów, które będą wyświetlane na ekranach komputerów, np. dokumentów w formacie PDF.

Pakiet `qtimes` umożliwia skład dokumentu w kroju `QTimes`, który jest klonem znanego kroju `Times New Roman` autorstwa Stanleya Morisona. Jeżeli dokument zawiera wzory matematyczne, to aby znaki w formułach były optycznie zgodne z otaczającym je tekstem, należy także dołączyć pakiety `txfonts` oraz `qtxmath`:

```
%& --translate-file=il2-pl
\documentclass[a4paper]{article}
\usepackage{polski}
\usepackage{txfonts,qtimes,qtxmath}
\usepackage{qswiss,qcourier}
\usepackage{sfheaders}
\author{Wanda Przechlewska}
\title{Test pakietu txfonts}
\begin{document} ...
```

Z kolei pakiet `qpalatin` wraz z pakietami `qpxmath` oraz `pxfonts` umożliwia skład dokumentu krojem `QPalatino` (łącznie ze wzorami matematycznymi), który jest klonem znanego kroju `Palatino` autorstwa Hermanna Zapfa:

```
%& --translate-file=il2-pl
\documentclass[a4paper]{article}
\usepackage{polski}
\usepackage{pxfonts,qpalatin,qpxmath}
\usepackage{qswiss,qcourier}
\usepackage{sfheaders}
\author{Maria Matysek}
\title{Test pakietu qpxfonts}
\begin{document} ...
```

W powyższych przykładach dołączono także pakiety `qswiss`, `sfheaders` oraz `qcourier`. Pakiet `qswiss` przestawia domyślną odmianę bezszeryfową na krój `Helvetica`, a `sfheaders` przeddefiniowuje śródtytuły tak, że składane są krojem bezszeryfowym. Z kolei `qcourier` zamienia domyślny krój o stałej szerokości znaków na `QCourier`. W rezultacie pierwszy z dokumentów będzie podobny

²Względna w porównaniu do nominalnego stopnia pisma; wysokość ta wynosi dla kroju CM 43%, a dla kroju `Palatino` – 46%; dla dużych liter z tych krojów jest to odpowiednio: 68,9% i 68,5%.

Rozdział 2

Składanie tekstu

Po lekturze poprzedniego rozdziału znasz już podstawowe pojęcia związane z systemem $\LaTeX 2_{\epsilon}$. W tym rozdziale Twoja wiedza wzbogaci się o informacje niezbędne do tworzenia prawdziwych dokumentów.

2.1. Struktura tekstu i języka

Hanspeter Schmid <hanspi@schmid-werren.ch>

Głównym zadaniem słowa pisanego jest przekaz myśli, informacji lub wiedzy. Nadanie zapisowi odpowiedniej struktury pomaga czytelnikowi lepiej zrozumieć przekazywane treści. Skład typograficzny może czytelnikowi tę logiczną i semantyczną strukturę tekstu przybliżyć.

\LaTeX tym się różni od innych systemów składu, że do złożenia tekstu wystarczy mu znajomość logicznej i semantycznej struktury tekstu. Postać typograficzna jest wyprowadzana na podstawie „reguł” zawartych w klasie dokumentu i plikach z makroinstrukcjami.

Najważniejszą jednostką podziału tekstu w \LaTeX u (a także w typografii) jest akapit. Jest to „jednostka podziału” głównie dlatego, że według wszelkich kanonów sztuki typograficznej treść akapitu powinna być poświęcona jednej spójnej myśli lub pojęciu. Tak więc, gdy zaczyna się nowa myśl, powinien się zacząć nowy akapit. Kontynuacja dotychczasowej myśli w nowym akapicie jest błędem. Niezgodne z regułami sztuki jest też pojawienie się w tym samym akapicie całkowicie nowej myśli. W następnym punkcie omówimy instrukcje nakazujące \LaTeX owi złamanie linii bez rozpoczynania nowego akapitu, na przykład poleceniem `\`, a także sposób rozpoczęcia nowego akapitu, na przykład przez umieszczenie w kodzie źródłowym pustej linii.

Większość ludzi całkowicie lekceważy znaczenie właściwej organizacji akapitów. Co więcej, wiele osób nawet nie zdaje sobie sprawy, czym akapit naprawdę jest, i (szczególnie w \LaTeX u) kończy akapit, nawet o tym nie wiedząc. Błąd taki łatwo popełnić zwłaszcza w tekście z równaniami.

Tabela 6.3: Wielkość stopnia pisma w klasach standardowych

Stopień	10pt	opcja 11pt	opcja 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

złoży w wyjściowym stopniu pisma. Jak widać, grupy można zagnieźdzać, nawet wielokrotnie.

Polecenia zmieniające stopień pisma zmieniają także interlinię. Dzieje się tak jednak tylko wtedy, gdy *przed* zamknięciem odpowiedniej grupy kończony jest akapit – przez wstawienie pustego wiersza *lub* polecenia `\par`. Zwróćmy uwagę na miejsce, w którym umieszczono instrukcję `\par` w poniższych dwóch przykładach.

```
{\Large Zdanie, które ma więcej
niż pięć słów, nie ma sensu!\par}
```

Zdanie, które ma więcej niż pięć słów, nie ma sensu!

```
{\Large Zdanie, które ma więcej
niż pięć słów, nie ma sensu!}\par}
```

Zdanie, które ma więcej niż pięć słów, nie ma sensu!

Jeśli zachodzi konieczność zmiany stopnia pisma dla całego akapitu lub jeszcze dłuższego tekstu, to możemy skorzystać ze składni przyjętej dla otoczeń:

```
\begin{Large} Zdanie, które ma
więcej niż pięć słów, nie ma sensu!
\end{Large}
```

Zdanie, które ma więcej niż pięć słów, nie ma sensu!

Zapis taki pozwala unikać łatwych do popełnienia błędów, wynikających z opuszczania nawiasów otwierających lub zamykających grupy.

W trybie matematycznym w celu złożenia fragmentu wzoru innym niż pochyłe krojem pisma można zastosować polecenia zestawione w tabeli 6.4.

wzięcia powietrza do płuc (lub zrobienia krótkiej przerwy), to postawmy tam przecinek. A najlepiej w celu rozwiania wątpliwości sięgnąć do słownika ortograficzno-interpunkcyjnego, który zresztą zawsze warto mieć pod ręką.

I na koniec: akapit nie jest największą logiczną jednostką podziału tekstu. Można jeszcze wspomnieć o rozdziałach, punktach, podpunktach itd. Jednakże od strony typograficznej już same nazwy poleceń w rodzaju `\section{Struktura tekstu i języka}` są na tyle oczywiste (dla znających język angielski), że sposób ich użycia jest łatwo zrozumiały.

2.2. Składanie akapitów i łamanie stron

2.2.1. Składanie akapitów

Książki najczęściej składa się tak, że wszystkie wiersze w akapitach są tej samej długości. Dążąc do optymalnej prezentacji akapitu, \LaTeX ustala miejsca złamań linii oraz wielkość odstępów między słowami. W razie potrzeby przenosi wyrazy, których nie jest w stanie zmieścić w wierszu. Sposób składania akapitów zależy od użytej klasy dokumentu. Najczęściej pierwszy wiersz akapitu jest wcięty, a między akapitami nie ma dodatkowych odstępów. Więcej na ten temat można przeczytać w punkcie 6.3.2.

Zgodnie z anglo-amerykańskimi zwyczajami typograficznymi \LaTeX nie wstawia wcięcia akapitowego bezpośrednio po tytułach rozdziałów, punktów itd. Polskie zwyczaje nakazują jednak rozpoczynanie także początkowych akapitów wcięciem. Efekt ten można osiągnąć przez dołączenie w preambule pakietu `indentfirst`. W niniejszym tłumaczeniu nie stosujemy wcięć w początkowych akapitach, gdyż akurat pod tym względem bardziej nam się podobają zwyczaje anglosaskie.

Czasami przydaje się instrukcja nakazująca \LaTeX owi złamanie linii. Polecenie:

```
\\ lub \newline
```

rozpoczyna nową linię bez rozpoczynania nowego akapitu. Natomiast instrukcja:

```
\\*
```

zakazuje dodatkowo złamania strony (w miejscu złamania linii). Z kolei instrukcja:

```
\newpage
```

rozpoczyna nową stronę. Instrukcje:

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



```
% Przykładowy pakiet ***
\ProvidesPackage{demopack}
\newcommand{\kwle}{Krótkie wprowadzenie do systemu \LaTeXe}
\newcommand{\wle}[1]{\emph{#1} wprowadzenie
do systemu \LaTeXe}
\newenvironment{zrodlo}{Źródło: }{\par}
```

Rysunek 6.1: Przykładowy pakiet

łączenia pakietu do dokumentu. Rysunek 6.1 przedstawia niewielki pakiet z instrukcjami z powyższych przykładów.

6.2. Fonty

6.2.1. Instrukcje przełączające stopień pisma

L^AT_EX automatycznie dobiera krój, odmianę i stopień pisma¹ dla różnych elementów dokumentu (tytułów rozdziałów, punktów, przypisów itp.). Czasami zachodzi jednak potrzeba „ręcznego” przełączenia kroju bądź stopnia pisma. Można do tego użyć poleceń zestawionych w tabelach 6.1 i 6.2. Stopień pisma jest kwestią układu graficznego dokumentu i zależy od wybranej klasy dokumentu oraz ustawienia odpowiednich opcji. W tabeli 6.3 zestawiono stopnie pisma w jednostkach absolutnych dla poleceń zmieniających wielkość kroju w standardowych klasach dokumentów.

```
{\small Nieliczni lecz
\textbf{odważni} Rzymianie rządzili}
{\Large wielką \textit{Italia}.}
```

Nieliczni lecz **odważni** Rzymianie rządzili
wielką *Italia*.

Biejący font jest w L^AT_EXu scharakteryzowany przez pięć elementów: układ (zestaw znaków), krój (rodzinę), grubość i szerokość, odmianę oraz stopień

¹*Pismo drukarskie* to pismo utrwalone na nośniku, tj. materiale, na którym umieszczono negatywy lub pozytywne znaki pisma. *Krój pisma* to obraz pisma drukarskiego o jednolitych cechach graficznych, niezależnych od stopnia i odmiany pisma. Każdy krój pisma posiada swoją nazwę (na przykład Times New Roman, Computer Modern czy Garamond). *Odmiana kroju* pisma różnicuje pisma jednego kroju ze względu na grubość, szerokość i pochylenie. *Stopień pisma* określa z kolei wielkość znaków. Zestawy metalowych *czcionek*, czyli kawałków metalu, w których utrwalano znaki pisma, drukarze przechowywali w *kasztach*. Angielską, a właściwie amerykańską nazwą zestawu czcionek jednego kroju i wielkości, powszechnie dziś używaną w terminologii komputerowej, jest *font* (zobacz też [3]).

Fonty L^AT_EXa, takie jak PL, EC czy LM, są optycznie identyczne, bo wszystkie są replikami kroju *Computer Modern*, różnią się jednak od najczęściej używanego w edytorach biurowych, takich jak MS Word, kroju *Times New Roman*.

między akapitami lub innymi elementami na stronie. Jeżeli wstawione odstępy okazują się za duże, co zdarza się dość często, to podczas przetwarzania dokumentu generowane jest odpowiednie ostrzeżenie, na przykład:

```
Underfull \vbox (badness 10000) has occurred
while \output is active [7]
```

W powyższym komunikacie (z uwagi na wąskość szpalty przełamanym na dwie linijki) liczba po słowie *badness* wskazuje, jak bardzo L^AT_EXowi nie podoba się złożona strona. Tutaj jest to 10000 – maksymalna w T_EXu ujemna ocena jakości składu. Liczba w nawiasach prostokątnych na końcu komunikatu oznacza numer strony, podczas składania której wystąpił problem.

L^AT_EX zawsze stara się znaleźć najlepszy podział akapitu na wiersze. Kiedy nie potrafi znaleźć podziału, który spełnia jego wysokie wymagania jakościowe, wtedy niektóre wyrazy wystają na prawy margines. Sytuacja taka jest sygnalizowana komunikatem podobnym do następującego:

```
Overfull \hbox (5.5452pt too wide) in paragraph at lines 79--83
```

W komunikacie tym liczba w nawiasie okrągłym oznacza, że pewien fragment tekstu wystaje o 5,5452 punktów drukarskich na prawy margines. Problem wystąpił w akapicie, który w pliku źródłowym jest w wierszach od 79 do 83. Podobne ostrzeżenia pojawiają się najczęściej wówczas, gdy L^AT_EX nie potrafi przenieść wyrazów w akapicie zgodnie z zadanymi wzorcami przenoszenia tak, by nie popsuć jakości składu. Komunikat typu *overfull hbox* nie wystarcza na ogół do dokładnego ustalenia przyczyny jego wystąpienia. Można wtedy jako argumentu polecenia `\documentclass` użyć opcji `draft`, na skutek czego L^AT_EX oznaczy wystające wiersze małą czarną sztabką na prawym marginesie szpalty.

Deklaracja `\sloppy` nakazuje L^AT_EXowi nieco obniżyć jego domyślnie wysokie standardy. Zapobiega to – w większości wypadków – występowaniu zbyt długich linijek, kosztem jednak zwiększenia odstępów międzywyrazowych, czyli pogorszenia jakości składu. Mogą się pojawiać ostrzeżenia typu *underfull hbox*, co w większości wypadków (zwłaszcza gdy podana miara kiejkości jest powyżej 5000) oznacza, że skład nie jest zbyt udany. Instrukcja `\fussy` działa w odwrotnym kierunku, to znaczy przywraca domyślne, wysokie standardy L^AT_EXa.

2.2.2. Przenoszenie wyrazów

W razie potrzeby L^AT_EX przenosi (dzieli) wyrazy. Jeżeli algorytm podziału przeniósł jakiś wyraz błędnie, to właściwe miejsca przenoszenia można zadać instrukcją:

```
\hyphenation{słowo1 słowo2 słowo3...}
```

Słowa z listy argumentów można dzielić wyłącznie w miejscach oznaczonych znakiem `-`. Instrukcji tej wolno użyć jedynie w preambule dokumentu,

Następny przykład ilustruje sposób wykorzystania opcjonalnego argumentu *num*. Znacznik #1 oznacza pierwszy parametr formalny (#2 oznaczałby drugi, #3 – trzeci itd.). W trakcie wykonywania treści instrukcji w miejsce parametrów formalnych T_EX wstawia argumenty podane w jej wywołaniu (parametry aktualne).

Instrukcja w poniższym przykładzie ma jeden parametr:

```
\newcommand{\wle}[1]
{\emph{#1} wprowadzenie
do systemu \LaTeXe}
% następnie po \begin{document}:
\wle{Krótkie}; \wle{Długie}
```

Krótkie wprowadzenie do systemu L^AT_EX 2_ε;
Długie wprowadzenie do systemu L^AT_EX 2_ε

L^AT_EX nie pozwala zdefiniować instrukcji nazwanej tak samo jak wcześniej zdefiniowane polecenie. W wypadku gdy chcemy zmienić znaczenie już istniejącej instrukcji, powinniśmy użyć polecenia `\renewcommand`. Za wyjątkiem nazwy ma ono składnię identyczną jak `\newcommand`.

Czasami może się też przydać polecenie `\providecommand`. Działa ono jak `\newcommand`, z tym że jeśli istnieje już komenda o takiej samej nazwie, to nie zastępuje ono starej, zwyczajnie ignorując nową.

Nieco uwagi należy poświęcić temu, jaki skutek ma umieszczenie odstępu po komendzie L^AT_EXa. Informacje na ten temat można znaleźć na stronie 5.

6.1.2. Otoczenia definiowane przez użytkownika

Odpowiednikiem definiującej nowe polecenie instrukcji `\newcommand` jest dla otoczeń instrukcja `\newenvironment`. Ma ona następującą składnię:

```
\newenvironment{nazwa}[num]{początek}{koniec}
```

Podobnie jak w wypadku `\newcommand`, można instrukcji `\newenvironment` użyć z argumentem opcjonalnym albo bez niego. L^AT_EX wstawia tekst *początek*, gdy w dokumencie napotyka napis `\begin{nazwa}`, a zawartość argumentu *koniec* – po napotkaniu napisu `\end{nazwa}`.

Poniższy przykład ilustruje sposób użycia instrukcji `\newenvironment`.

```
\newenvironment{zrodlo}
{Źródło: }{\par}
% następnie po \begin{document}:
\begin{zrodlo} Rocznik Statystyczny
GUS. \end{zrodlo}
```

Źródło: Rocznik Statystyczny GUS.

Znaczenie argumentu *num* jest takie samo jak w instrukcji `\newcommand`.

L^AT_EX nie pozwala zdefiniować otoczenia o już istniejącej nazwie. W razie potrzeby zastąpienia już istniejącego otoczenia powinniśmy użyć polecenia `\renewenvironment`, o składni takiej samej jak `\newenvironment`.

Instrukcja	Przykład	Opis
<code>\today</code>	21 stycznia 2007	Bieżąca data
<code>\TeX</code>	T _E X	Twój ulubiony system składni
<code>\LaTeX</code>	L ^A T _E X	Nazwa tej gry
<code>\LaTeXe</code>	L ^A T _E X 2 _ε	Obecne jej wcielenie

2.4. Znaki specjalne i symbole

2.4.1. Cudzysłowy

Znaku cudzysłowu " używa się inaczej niż na maszynie do pisania. W publikacjach drukowanych różnie oznacza się początek i koniec cudzysłowu. Występują także różnice w sposobach oznaczania cudzysłowów w różnych językach. Dwa apostrofy ‘ otwierają, a dwa apostrofy ’ zamykają L^AT_EXowy cudzysłów według reguł języka angielskiego:

‘Please press the ‘x’ key.’

“Please press the ‘x’ key.”

W języku polskim cudzysłów otwierający oznacza się dwoma przecinkami ,, , natomiast zamykający – dwoma apostrofami ’ ’³. Gdy zachodzi konieczność użycia cudzysłowu w tekście już objętym cudzysłowem, to stosuje się „cudzysłowy «francuskie»”, oznaczane w pliku źródłowym znakami, odpowiednio, mniejszości << i większości >>.

„Przechodź tylko po <<zebrach>>”!

„Przechodź tylko po «zebrach»”!

2.4.2. Pauzy i myślniki

Zwyczajne maszyny do pisania posiadają tylko jeden znak w kształcie poziomej kreski „-”. W składzie drukarskim występują aż cztery rodzaje kresek poziomych. Są to: łącznik, myślnik, półpauza i minus, używany we wzorach matematycznych.

Łącznik (dywiz) jest najkrótszą z kresek. Stosuje się go do dzielenia i przenoszenia wyrazów oraz do łączenia wyrazów wielocłonowych (np. „niebiesko-czarny”). Zgodnie z polskimi regułami wyraz wielocłonowy można podzielić i przenieść albo w obrębie wyrazów składowych, albo na łączniku. W drugim z tych przypadków łącznik należy powtórzyć, to znaczy powinien się on znaleźć zarówno na końcu pierwszego wiersza, jak też na początku drugiego. Oto możliwe miejsca podziału wyrazu niebiesko-czarny:

³Porównaj uwagi o tym sposobie oznaczania cudzysłowów z punktu 2.5 w części dotyczącej fontów i ich kodowania (str. 27). Dotyczy to również cudzysłowów francuskich.

Rozdział 6

Adaptowanie L^AT_EXa

Dokumenty składane za pomocą poznanych do tej pory poleceń będą się zapewne podobały zdecydowanej większości czytelników. Chociaż ich wygląd nie będzie może wyrafinowany, z pewnością jednak spełnią one podstawowe zasady składu, dzięki czemu będzie się je czytało łatwo i przyjemnie.

W niektórych wypadkach może się jednak okazać, że brakuje polecenia czy otoczenia, za pomocą którego moglibyśmy złożyć dany fragment tekstu w sposób odpowiadający potrzebom, albo też że sposób działania dostępnej w L^AT_EXu instrukcji nie spełnia naszych wymagań.

W tym rozdziale przedstawimy, jak nauczyć L^AT_EXa formatować dokumenty tak, aby wyglądały inaczej niż w wypadku korzystania jedynie ze standardowych klas i pakietów.

6.1. Definiowane instrukcji i otoczeń

Czytelnicy zauważyli zapewne, że nowo wprowadzane w tej książce polecenia ukazują się w ramkach oraz że znajdują się one w skorowidzu. Aby to osiągnąć, nie korzystaliśmy za każdym razem z wbudowanych w L^AT_EXa instrukcji, lecz utworzyliśmy własny pakiet, w którym zawariliśmy nowe, potrzebne nam polecenia i otoczenia. Dysponując takim pakietem, wystarczy po prostu napisać:

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

```
\polecenie
```

W tym przykładzie użyliśmy zarówno nowego otoczenia o nazwie `command`, odpowiedzialnego za rysowanie ramek dookoła instrukcji, jak też nowego polecenia `\ci`, służącego do składu nazw poleceń i wprowadzania ich do skorowidza. Proponujemy Czytelnikom odszukanie hasła `\polecenie` w skorowidzu; przy hasle powinny być podane numery stron, na których ta instrukcja występuje w książce.

godz.~17.00; od~15 do 40~osób; na str.~2 napisano; rozdz.~2;
2~rozdziały; p.~Jan Nowak; p.~J.~Nowak; I~część IX~Symfonii.
Ze względu na zależność od kontekstu obowiązek decydowania o użyciu tyldy spada na użytkownika L^AT_EXa.

2.4.4. Tylda (~)

W adresach internetowych często występuje znak tyldy. W L^AT_EXu można by do jego uzyskania użyć instrukcji `\~`, ale wynik: `~` nie jest chyba tym, czego oczekujemy. Lepiej zrobić tak:

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

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

A jeszcze lepiej dołączyć do preambuły dokumentu pakiet `url` i korzystać z polecenia `\url{...}`.

2.4.5. Oznaczenie stopni (°)

Poniższy przykład ilustruje, jak w L^AT_EXu uzyskuje się symbol stopni:

```
Jest $-30\,^{\circ}\mathrm{C}$.
Niedługo zacznę nadprzewodzić.
```

```
Jest -30°C. Niedługo zacznę nadprzewodzić.
```

Pakiet `textcomp` udostępnia symbol stopni także jako `\textcelsius`.

2.4.6. Symbol waluty euro (€)

Pisząc dziś o pieniądzach, nie można się obejść bez symbolu euro. Znak ten występuje w wielu współczesnych fontach. Po załadowaniu pakietu `textcomp` w preambule:

```
\usepackage{textcomp}
```

można do uzyskania symbolu euro użyć oznaczenia:

```
\texteuro
```

Jeśli używany font nie zawiera własnego symbolu euro albo nam się on nie podoba, to mamy dwie dodatkowe możliwości: Pierwszą jest pakiet `eurosym`. Udostępnia on oficjalny znak euro:

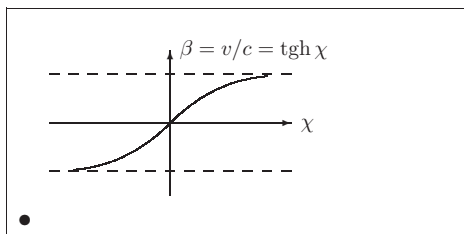
```
\usepackage[official]{eurosym}
```

Jeśli wolimy znak euro zgodny optycznie z fontem, to zastąpmy opcję `official` opcją `gen`.

jego lewemu dolnemu narożnikowi (oznaczonemu czarnym kółeczkiem) przypisano współrzędne $(-2.5, -0.25)$.

5.2.11. Prędkość w Szczególnej Teorii Względności

```
\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}
```



Punkty kontrolne dwóch krzywych Béziera wyliczono ze wzorów (5.1). Gałąź dodatnia jest określona wartościami $P_1 = (0, 0)$, $m_1 = 1$ oraz $P_2 = (2, \tanh 2)$, $m_2 = 1/\cosh^2 2$. I znowu rysunek wyrażono w wygodnych matematycznie współrzędnych, a lewemu dolnemu narożnikowi przypisano współrzędne $(-3, -2)$ (czarne kółeczko).

5.3. Xy-pic

Alberto Manuel Brandão Simões <albie@alfarrabio.di.uminho.pt>

Pakiet Xy-pic służy do rysowania grafów. Aby uzyskać dostęp do jego funkcji, umieszczamy w preambule dokumentu wiersz:

```
\usepackage[opcje]{xy}
```

Parametr *opcje* jest listą funkcji pakietu Xy-pic, które mają zostać załadowane. Opcje te przydają się między innymi do szukania błędów w pakiecie. Zaleca się przekazywać opcję `all`, nakazującą L^AT_EXowi załadować wszystkie polecenia pakietu.

Graf rysuje się w Xy-pic na płótnie o strukturze macierzy, a każdy element grafu jest umieszczony w komórce tej macierzy:

o rozbudowanej strukturze, na przykład zawierające skorowidze. Inne podejście do sprawy języka polskiego w L^AT_EXu jest przedstawione w punkcie 2.5.1.

Kodowanie znaków ma znaczenie nie tylko w pliku źródłowym. Drugą stroną medalu jest układ znaków w foncie, czyli kodowanie fontu. Określa ono, w których miejscach *fontu* znajdują się poszczególne znaki. Standardowo L^AT_EX używa kodowania o nazwie OT1, przyjętego dla oryginalnych T_EXowych fontów Computer Modern (CM). Są to fonty jedynie 128-znakowe, nie zawierają na przykład charakterystycznych polskich liter. Znak diakrytyczny można w nich skonstruować metodą nałożenia dwóch innych: litery i odpowiedniego akcentu. Ta metoda ma wady, bo T_EX nie może poprawnie przenieść wyrazów zawierających tak zapisane znaki diakrytyczne, kiepska jest też jakość typograficzna takich diakrytyków.

Na szczęście wszystkie współczesne dystrybucje T_EXa zawierają komplet fontów *European Computer Modern* (EC). Są to fonty zawierające do 256 znaków w kodowaniu T1. Pierwszych 128 znaków fontu EC jest (prawie) identycznych jak w odpowiadającym mu foncie CM. Pozostałe 128 znaków to znaki diakrytyczne występujące w różnych językach europejskich, w tym też komplet znaków niezbędnych do składania tekstów polskich. Fonty EC umożliwiają poprawne przenoszenie wyrazów, znacznie lepsza jest też jakość typograficzna znaków diakrytycznych.

Polscy użytkownicy mogą też korzystać z rodziny fontów PL (autorzy B. Jackowski, M. Ryćko, J. Nowacki) oraz nowszych fontów LM (autorzy B. Jackowski i J. Nowacki). W zakresie objętym przez rodzinę CM fonty PL/LM są całkowicie z nią zgodne, a dodatkowo zawierają wszystkie polskie znaki diakrytyczne. Polskie diakrytyki w fontach EC są kopią odpowiednich znaków z fontów PL, czyli „ogonki” wyglądają tak samo, niezależnie od tego, z której z rodzin korzystamy.

W fontach EC, PL i LM dostępne są ponadto znaki cudzysłowów francuskich i polskiego otwierającego, których nie ma w fontach CM. Opisany w punkcie 2.4.1 sposób wprowadzania tych znaków za pomocą par `, , <<` i `>>` działa jedynie wówczas, gdy używamy fontów EC, PL bądź LM.

Aby przełączyć się na fonty EC lub PL, trzeba do preambuły dokumentu dołączyć pakiet `fontenc`:

```
\usepackage[T1]{fontenc} lub \usepackage[OT4]{fontenc}
```

Argument T1, określający kodowanie, jest „odpowiedzialny” za przełączenie się na fonty EC. Podobnie Argument OT4 włącza fonty PL.

Uwaga: Powyższe dołączenie pakietu `fontenc` jest jedynie deklaracją. Jeżeli twoja dystrybucja L^AT_EXa nie zawiera fontów, które chcesz włączyć, to powyższe polecenia nie zostaną wykonane, a L^AT_EX przełączy się na font domyślny, zwykle nie zawierający polskich znaków.

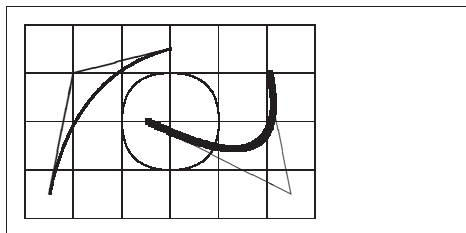
Reasumując, oto przykładowa preambuła artykułu składanego w języku polskim przy wykorzystaniu pakietów `babel`, `inputenc` oraz `fontenc`:

Argument *nazwa* odnosi się do L^AT_EXowego magazynu poleceń i dlatego ma naturę instrukcji (co w powyższym przykładzie objawia się choćby użyciem znaków `\`). Rysunki przechowywane w pudełkach można zagnieżdżać. W powyższym przykładzie wewnątrz definicji `\folderb` użyto `\foldera`.

Trzeba było użyć polecenia `\oval`, ponieważ instrukcja `\line` nie działa, gdy długość odcinka wynosi mniej niż około 3mm.

5.2.9. Krzywe Béziera drugiego stopnia

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



Jak widać w przykładzie, podzielenie okręgu na cztery krzywe Béziera drugiego stopnia nie daje zadowalającego efektu; lepsze przybliżenie dałoby osiem. Przykład ponownie ilustruje wpływ instrukcji `\linethickness` na linie poziome i pionowe oraz poleceń `\thinlines` i `\thicklines` na odcinki pochyłe. Pokazuje on również, że oba te rodzaje poleceń oddziałują na krzywe Béziera i że kolejne użycie któregośkolwiek z nich przesłania poprzednie.

Niech symbole $P_1 = (x_1, y_1)$, $P_2 = (x_2, y_2)$ oznaczają punkty końcowe, zaś m_1, m_2 – odpowiednie nachylenia krzywej Béziera drugiego stopnia. Pośredni punkt kontrolny $S = (x, y)$ jest zatem opisany równaniami:

$$\begin{cases} x = \frac{m_2 x_2 - m_1 x_1 - (y_2 - y_1)}{m_2 - m_1}, \\ y = y_i + m_i(x - x_i) \quad (i = 1, 2). \end{cases} \quad (5.1)$$

```
\k{a} \ 'c \k{e} \l{l} \ 'n \ 'o \ 's
\ 'z \ .z \k{A} \ 'C \k{E} \L{L} \ 'N
\ 'O \ 'S \ 'Z \ .Z
```

ą ć ę ł ń ó ś ź ż Ą Ć Ę Ł Ń Ó Ś Ź Ż

Posługiwanie się powyższymi poleceniami do pisania tekstów po polsku jest uciążliwe, ale przydaje się na przykład do wstawienia niewielkich fragmentów do dokumentu, który będzie przetwarzany przez kogoś, kto języka polskiego nie zna i nie ma dobrze skonfigurowanej pod tym kątem instalacji (np. polskie wstawki w artykule konferencyjnym pisanym po angielsku).

Bezpośrednie wprowadzanie polskich znaków umożliwia mechanizm przekodowywania, uruchamiany przez umieszczenie komentarza strukturalnego w pierwszym wierszu pliku⁸:

```
%& --translate-file=cp1250pl
```

Wpis taki będzie poprawny w wypadku redagowania plików w systemie MS Windows i stosowania domyślnego w nim kodowania CP 1250. W wypadku systemu Unix/Linux należy zamiast `cp1250pl` wpisać `il2-pl` (przy założeniu, że posługujemy się kodowaniem ISO 8859-2).

Komentarz strukturalny jest alternatywą dla pakietu `inputenc`, opisanego w punkcie 2.5⁹. Mechanizm komentarza strukturalnego nie umożliwia poprawnego przetwarzania dokumentów unikodowych. Jeżeli upierasz się przy unikodzie to pozostaje ci tylko pakiet `inputenc`.

Zamiast pakietu `babel` można zastosować pakiet polski z zestawu `platex` (autorzy Mariusz Olko i Marcin Woliński). Jego niewątpliwą zaletą jest starsza polonizacja, np. pakiet polski domyślnie przełącza się na fonty PL bez potrzeby dołączania pakietu `fontenc`. W wypadku przejścia z pakietu `babel` na polski dokumenty nie wymagają modyfikacji, oprócz oczywistej wymiany nazwy ładowanego pakietu w preambule.

Pakiet polski dołączamy w preambule dokumentu poleceniem:

```
\usepackage[opcje]{polski}
```

Lista ważniejszych opcji obejmuje:

- OT4 wybranie kodowania OT4 fontów, co w praktyce oznacza skład fontami PL;
- T1 wybranie kodowania T1 fontów, co oznacza skład fontami EC;
- OT1 wybranie kodowania OT1 fontów, co oznacza skład fontami CM (niezalecane);
- `plmath` włączenie polskich oznaczeń dla standardowych poleceń trygonometrycznych oraz symboli relacji mniejszy-lub-równy i większy-lub-równy (zalecane);

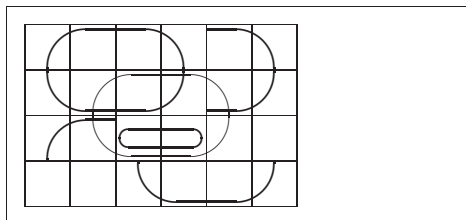
⁸Komentarz ten należy umieścić w pierwszym wierszu pliku, czyli jeszcze przed wierszem z `\documentclass`, a znak `%` musi być pierwszym znakiem tego wiersza.

⁹Oznacza to, że jednoczesne użycie komentarza strukturalnego `translate-file` i dołączanie pakietu `inputenc` jest błędem.

`\linethickness` odnosi się do odcinków poziomych oraz pionowych, jednakże nie do odcinków ukośnych ani nie do okręgów. Stosuje się ona jednak również do krzywych Béziera drugiego stopnia.

5.2.7. Owale

```
\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)[t]}
\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}
```



Polecenie:

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

jak też:

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

generuje owal o środku (x, y) , szerokości w i wysokości h . Opcjonalne argumenty *pozycji*: **b**, **t**, **l** i **r** oznaczają odpowiednio: „top” (góra), „bottom” (dół), „left” (lewo) i „right” (prawo). Jak pokazuje przykład, można także używać ich kombinacji.

Grubością linii można sterować na dwa sposoby, z jednej strony poleceniem:

```
\linethickness{length}
```

z drugiej – za pomocą `\thinlines` i `\thicklines`. O ile pierwszy ze sposobów odnosi się jedynie do linii poziomych oraz pionowych, a także do krzywych Béziera drugiego stopnia, o tyle `\thinlines` i `\thicklines` stosują się do odcinków ukośnych, jak też do okręgów i owali.

Jeżeli bezpośrednio przed kropką znajduje się duża litera, to \LaTeX nie traktuje takiego miejsca jako końca zdania, lecz jako kropkę po skrótce.

Wyjątki od powyższych zasad trzeba wyraźnie zaznaczyć w tekście. Znak `\` poprzedzający spację oznacza odstęp normalnej wielkości. Tylda `~` również wstawia taki odstęp, z tym że \LaTeX owi nie wolno na nim złamać wiersza. Umieszczenie instrukcji `\@` przed kropką jest dla \LaTeX a wskazówką, że ta kropka kończy zdanie, nawet jeśli następuje po dużej literze.

```
Pan~Kowalski ucieszył się\\
na jej widok (zob.~Rys.~5).\\
Podoba mi się JAVA\@. A~tobie?
```

```
Pan Kowalski ucieszył się
na jej widok (zob. Rys. 5).
Podoba mi się JAVA. A tobie?
```

Jak wspomniano, wstawianie większych odstępów na końcu zdań to zwyczaj anglosaski. W Europie kontynentalnej tradycyjnie się tego nie robi. Wstawianie większych odstępów na końcu zdań można wyłączyć poleceniem:

```
\frenchspacing
```

Pakiet polski domyślnie wykonuje instrukcję `\frenchspacing` za nas, czyli włącza odstęp „kontynentalny”.

2.7. Tytuły, śródtytuły i punkty

Podzielenie dokumentu na rozdziały, punkty, podpunkty itd. pomaga czytelnikom lepiej orientować się w tekście. Do dzielenia dokumentu na hierarchiczne części służą odpowiednie instrukcje \LaTeX owe. Do autora należy używanie tych poleceń w odpowiednim porządku.

W klasie `article` mamy do dyspozycji następujące instrukcje podziału hierarchicznego:

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

W klasach `report` (raport) i `book` (książka) mogą występować rozdziały: `\chapter{...}`

Jeśli raport bądź książkę trzeba podzielić na części bez naruszania numeracji punktów i rozdziałów, to można użyć polecenia:

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

Ponieważ w klasie `article` najwyższą jednostką w hierarchii podziału jest `\section` (czyli *punkt*), łatwo tworzy się książkę (klasa `book`), w których rozdziałami są poszczególne artykuły. \LaTeX dobierze za nas odpowiednie odstęp między rozdziałami oraz wielkość i krój pisma w śródtytułach.

Dwie z wymienionych instrukcji działają nieco inaczej niż pozostałe:

5.2.4. Okręgi

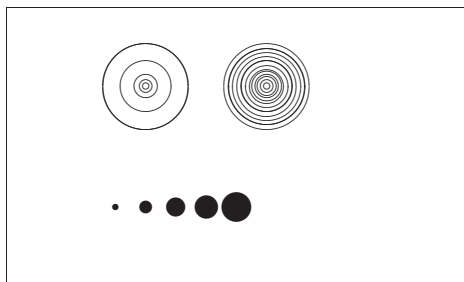
```

\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}

```



Polecenie:

```
\put(x,y){\circle{diam}}
```

rysuje okrąg o środku (x, y) i średnicy (nie promieniu) równym $diam$. Otoczenie `picture` dopuszcza średnice co najwyżej rzędu 14 mm, na dodatek nie wszystkie długości średnic są dozwolone. Polecenie `\circle*` generuje koła (wypełnione okręgi).

Podobnie jak w wypadku odcinków, można się uciec do dodatkowych pakietów, takich jak `eeic` bądź `pstricks`. Obszerny opis tych pakietów można znaleźć w [7].

Otoczenie `picture` oferuje jeszcze jedną opcję. Jeśli nie boisz się wykonania niezbędnych obliczeń (być może nawet za pomocą programu), to okręgi oraz elipsy możesz złatać z krzywych Béziera drugiego stopnia. Przykłady oraz źródłowe pliki w Javie można znaleźć w [21].

```

\chapter*{Wstęp}
\addcontentsline{toc}{chapter}{Wstęp}

```

Najczęściej hasła w spisie treści pokrywają się z tytułami rozdziałów czy punktów. Czasami jednak nie jest to pożądane, na przykład wówczas, gdy tekst hasła jest zbyt długi. W takich wypadkach hasło do spisu treści można podać jako *opcjonalny* argument instrukcji podziału hierarchicznego, na przykład tak:

```

\chapter[Krótki i~ekscytujący rozdział]{To jest
bardzo długi i~wyjątkowo nudny rozdział}

```

W wyniku wykonania tej instrukcji w spisie treści pojawi się „Krótki i ekscytujący rozdział”, natomiast w tytule rozdziału „To jest bardzo długi i wyjątkowo nudny rozdział”.

L^AT_EX składa część tytułową dokumentu, napotkawszy instrukcję:

```
\maketitle
```

Należy ją umieścić po `\begin{document}`, czyli nie w preambule.

Zawartość części tytułowej ustalają polecenia:

```
\title{...}, \author{...} oraz opcjonalnie \date{...}
```

Należy je umieścić w preambule. Jeżeli dokument ma kilku autorów, to ich nazwiska i imiona rozdzielamy instrukcją `\and`. Sposób użycia powyższych instrukcji demonstruje rysunek 1.2 ze strony 8.

W L^AT_EX 2_ε istnieją trzy dodatkowe instrukcje dotyczące struktury dokumentu, dostępne jednak wyłącznie w klasie `book`. Oto ich nazwy, sposób użycia i przeznaczenie:

`\frontmatter` powinna być pierwszą instrukcją w treści dokumentu, czyli powinna wystąpić tuż po `\begin{document}`. Włącza ona rzymski zapis numerów stron, wyłączając jednocześnie numerowanie punktów podziału. To tak, jakby się używało instrukcji „gwiazdkowanych”, w rodzaju `\chapter*{Preface}`. Punkty podziału trafiają jednak do spisu treści.

`\mainmatter` należy umieścić tuż przed pierwszym rozdziałem książki. Przełącza ona sposób oznaczania numerów stron na arabski, zerując zarazem licznik stron.

`\backmatter` powinna wystąpić przed ostatnimi fragmentami książki, takim jak spis literatury albo skorowidz.

Powyższe instrukcje przydają się do podzielenia książki na część wstępną (obejmującą tytulaturę, spisy treści, tabel, wstępy itd.), główną i zakończenie (załączniki, skorowidze, kolofon itd.). W części wstępnej tytuły rozdziałów są składane mniejszym stopniem pisma (czego należy oczekiwać), a numery stron są w notacji rzymskiej (co raczej odbiega od polskich zwyczajów typograficznych).

Wartością domyślną `\unitlength` jest 1pt. Pierwsza para, (x, y) , to wymiary rezerwowanego wewnątrz dokumentu prostokątnego obszaru na rysunek. Opcjonalna druga para, (x_0, y_0) , to współrzędne przypisane dolnemu lewemu narożnikowi zarezerwowanego prostokąta.

Większość poleceń rysujących ma jedną z dwóch postaci:

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

lub

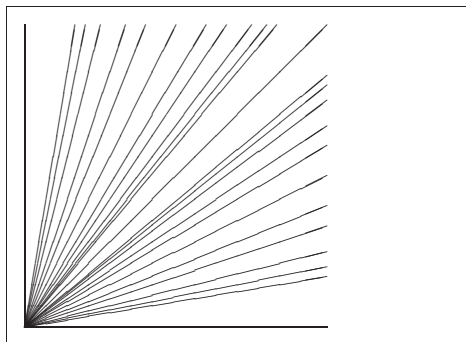
```
\multiput(x,y)(\Delta x,\Delta y){n}{object}
```

Wyjątkiem są krzywe Béziera, gdyż rysuje się je poleceniem:

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

5.2.2. Odcinki

```
\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}
```



Przypisy`\footnote{To jest właśnie przypis.}` są często stosowane przez użytkowników `\LaTeX`a

Przypisy^a są często stosowane przez użytkowników `\LaTeX`a

^aTo jest właśnie przypis.

2.10. Wyróżnienia

W tekstach pisanych na maszynie fragmenty, które mają zostać wyróżnione, podkreśla się. W dokumentach drukowanych wyróżnienie fragmentu odbywa się przez złożenie go *kursywą*. Służy do tego `\LaTeX`owa instrukcja:

```
\emph{tekst}
```

Argumentem tej instrukcji jest tekst, który ma zostać wyróżniony.

```
\emph{\emph{Wyróżnienia} w tekście
już wyróżnionym są składane
\emph{zwykłym} krojem pisma.}
```

Wyróżnienia *w tekście już wyróżnionym są składane zwykłym krojem pisma.*

Zwróćmy uwagę, że istnieje różnica między wyróżnieniem części tekstu a złożeniem go inną czcionką:

```
\textit{Tekst można \emph{wyróżnić},
składając go kursywą,}
\textsf{czcionką \emph{szeryfową},}
\texttt{a nawet \emph{maszynowo}.}
```

Tekst można wyróżnić, składając go kursywą, czcionką szeryfową, a nawet maszynowo.

2.11. Otoczenia

Do instrukcji formatujących zaliczają się *otoczenia* (zwane też *środowiskami*), czyli instrukcje postaci:

```
\begin{nazwa} tekst \end{nazwa}
```

gdzie *nazwa* jest nazwą otoczenia, a *tekst* jest fragmentem dokumentu, który ma zostać złożony inaczej niż poza otoczeniem.

Otoczenia można zagnieżdżać jedno w drugim:

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

Niedopuszczalne jest natomiast „przeplatanie” otoczeń:

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

Wiele standardowych poleceń `\LaTeX`a można zapisać w formie „otoczeniowej”. W takich wypadkach nazwa polecenia (bez w-tył-ciacha) jest nazwą otoczenia. Na przykład, zamiast polecenia `\em`, włączającego wyróżniający krój pisma, możemy zastosować otoczenie `\begin{em}... \end{em}`.

W kolejnych punktach przedstawiamy częściej używane otoczenia.

Rozdział 5

Tworzenie grafiki matematycznej

Większość ludzi używa \LaTeX a do składania tekstów. Ponieważ jednak podejście strukturalno-logiczne do tworzenia dokumentów jest tak wygodne, \LaTeX oferuje pewną – fakt, że obarczoną ograniczeniami – możliwość generowania grafiki z opisów tekstowych w pliku źródłowym. Co więcej, powstało sporo rozszerzeń \LaTeX a przełamujących wspomniane ograniczenia. W tym rozdziale dowiesz się o kilku z tych rozszerzeń.

5.1. Przegląd

Otoczenie `picture` pozwala programować rysunki bezpośrednio w \LaTeX u. Szczegółowy jego opis można znaleźć w [12]. Metoda ta ma jednak ograniczenia, które wynikają z tego, że zarówno nachylenia odcinków, jak i średnice okręgów można wybierać jedynie spośród niewielkiej liczby wartości. Z drugiej strony w wersji \LaTeX 2 ϵ można w otoczeniu `picture` używać polecenia `\qbezier`, gdzie „q” oznacza drugiego stopnia (ang. *quadratic*). Wiele często używanych krzywych, jak: okręgi, elipsy albo krzywe łańcuchowe można – niekiedy z odrobiną wysiłku matematycznego – zadowalająco przybliżyć krzywymi Béziera¹. Jeśli na dodatek do generowania \LaTeX owych bloków `\qbezier` użyje się języka programowania, na przykład Javy, to otoczenie `picture` ujawni całkiem pokazną moc.

Chociaż programowanie rysunków w \LaTeX u wiąże się z ograniczeniami i jest niekiedy nużące, to jednak są powody, by z tej możliwości korzystać. Dokumenty wytworzone w ten sposób są „małe” – w sensie liczby bajtów zajmowanych przez wynikowe pliki PDF lub DVI – a na dodatek nie trzeba do nich wczytywać dodatkowych plików graficznych.

¹W ciekawym artykule [10] B.L. Jackowski zwraca uwagę, że krzywe Béziera są krzywymi giętymi stopnia trzeciego, nie zaś – jak sugeruje nazewnictwo \LaTeX owe – stopnia drugiego.

2.11.3. Otoczenia `quote`, `quotation` i `verse`

Otoczenie `quote` przydaje się do składania cytatów oraz przykładów:

Jeżeli chodzi o długość wierszy, to regułą kciuka jest, że:
`\begin{quote}`
 Przeciętnie wiersz nie powinien zawierać więcej niż 66 znaków.

Dlatego w \LaTeX u standardowe strony mają szerokie marginesy.
`\end{quote}`
 Dlatego też w gazetach stosuje się druk wielołamowy.

Jeżeli chodzi o długość wierszy, to regułą kciuka jest, że:
 Przeciętnie wiersz nie powinien zawierać więcej niż 66 znaków.
 Dlatego w \LaTeX u standardowe strony mają szerokie marginesy.
 Dlatego też w gazetach stosuje się druk wielołamowy.

Istnieją ponadto dwa otoczenia o podobnym zastosowaniu: `quotation` oraz `verse`. Pierwsze z nich przydaje się do formatowania cytatów dłuższych niż jeden akapit. W przeciwieństwie do otoczenia `quote`, wewnątrz `quotation` \LaTeX rozpoczyna poszczególne akapity od wcięcia akapitowego. Otoczenie `verse` służy do składania wierszy. Poszczególne linijki zwrotek należy kończyć instrukcją `\\`, poszczególne zaś zwrotki – oddzielać pustą linią.

Na pamięć znam tylko jeden angielski wiersz. Ten 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}`

Na pamięć znam tylko jeden angielski wiersz. Ten 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. Streszczenie

Publikacje naukowe zaczynają się zazwyczaj od streszczenia – przeglądu tego, co czytelnik napotka w dalszej części. W \LaTeX u do wyróżniania streszczeń służy otoczenie `abstract`. Używa się go na ogół w dokumentach klasy `article`.

`\begin{abstract}`
 Streszczenie streszczenia.
`\end{abstract}`

Streszczenie streszczenia.

```
%& --translate-file=il2-pl
\documentclass[10pt]{beamer}
\mode<beamer>{\usetheme[hideothersubsections,%
  right,width=22mm]{Goettingen} }
\usepackage{polski}
\title{Prosta prezentacja}
\author[K. Wielki]{Karol Wielki}
\institute{Uniwersytet Karola Wielkiego}
%\titlegraphic{\includegraphics[width=20mm]{USTL}}
\date{2005}

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

\section{Przykład}

\begin{frame}
  \frametitle{Co robić w niedzielne popołudnie?}
  \begin{block}{Można:}
    \begin{itemize}
      \item pójść na spacer z psem\pause
      \item przeczytać książkę\pause
      \item pobawić się z kotem\pause
    \end{itemize}
  \end{block}
  i zrobić wiele innych rzeczy.
\end{frame}
\end{document}
```

Rysunek 4.2: Prosty kod dla klasy beamer

globalną, wpisując na przykład `\documentclass[10pt,handout]{beamer}` w celu wydrukowania materiałów do rozdania.

Wygląd ekranu prezentacji zależy od wybranego *tematu*. Możemy wskazać jeden z tematów dostarczanych wraz z klasą albo stworzyć nasz własny. Więcej informacji na ten temat można znaleźć w opisie klasy zawartym w pliku `beameruserguide.pdf`.

Przyjrzyjmy się bliżej kodowi z rysunku 4.2. Do wersji ekranowej prezentacji wybrano dla trybu `\mode<beamer>` temat *Goettingen*, w którym spisowi treści towarzyszy panel do nawigacji. Opcje tematu pozwalają określić rozmiar panelu (w tym wypadku 22mm) oraz jego pozycję (z prawej strony głównego tekstu). Opcja `hideothersubsections` nakazuje pokazywać tytuły punktów, jednak tylko te, które pochodzą z bieżącego rozdziału. Dla

Dla każdej kolumny należy w argumentcie wstawić jedną z liter: `l`, `r` lub `c`, określając w ten sposób justowania zawartości kolumny. Dosunięcie zawartości kolumny do lewej oznaczamy literą `l`, do prawej – znakiem `r`, a wyśrodkowanie – znakiem `c`. Zapisu `p{szer-kolumn}` można użyć do zaznaczenia, że kolumna ma mieć szerokość *szer-kolumn*. Wewnątrz takiej kolumny tekst jest składany w prostokąt o zadanej szerokości, z wyrównywaniem obu marginesów. Znak `|` instruuje \LaTeX a, by kolumny tabeli rozdzielił pionową kreską.

Wewnątrz otoczenia `tabular` poszczególne wiersze oddzielamy instrukcją `\\`, a rubryki w wierszu – znakiem `&`. Instrukcja `\hline` wstawia poziomą kreskę na całą szerokość tabeli.

```
\begin{tabular}{|r|l|} \hline
7C0 & heksadecymalnie \\
3700 & oktalnie \\
11111000000 & binarnie \\
\hline \hline
1984 & dziesiętnie \\
\end{tabular}
```

7C0	heksadecymalnie
3700	oktalnie
11111000000	binarnie
1984	dziesiętnie

```
\begin{tabular}{|p{4.7cm}|} \hline
Ten akapit jest wewnątrz pudełka.
Mamy nadzieję, że uzyskany
efekt się podoba.\\ \hline
\end{tabular}
```

Ten akapit jest wewnątrz pudełka. Mamy nadzieję, że uzyskany efekt się podoba.
--

Instrukcją `@{...}` określamy odstęp między kolumnami. Zastępuje ona domyślny odstęp międzykolumnowy treścią umieszczoną między `{ a }`. Stosuje się ją często do wyrównywania zestawień liczbowych według cyfr znaczących. Można ją także wykorzystać do usunięcia odstępów w pierwszej i ostatniej kolumnie tabeli, co ilustruje poniższy przykład:

```
\begin{tabular}{@{} l @{} } \hline
bez odstępów na brzegach\\ \hline
\end{tabular}
```

bez odstępów na brzegach

```
\begin{tabular}{|l} \hline
odstęp na brzegach tabeli\\ \hline
\end{tabular}
```

odstęp na brzegach tabeli

W \LaTeX u nie ma mechanizmu pozwalającego wyrównywać zestawienia liczbowe według cyfr znaczących¹¹, ale efekt ten można uzyskać, składając liczbę w dwóch kolumnach: część całkowitą w kolumnie wyrównywanej do prawego brzegu i część dziesiętną w kolumnie wyrównywanej do lewego. Za pomocą instrukcji `@{,}` zastępujemy przecinkiem odstęp wstawiany normalnie między kolumnami. Trzeba jednak pamiętać o konieczności wpisywania znaku `&` zamiast przecinków w liczbach. Rubryki rozciągające się na kilka kolumn, jak nagłówki w poniższym przykładzie, tworzymy poleceniem `\multicolumn`:

¹¹Do wyrównywania cyfr można skorzystać z pakietu `dcolumn` z zestawu „tools”.

Na zakładce otrzymamy „redRed!”. Polecenie `\textcolor` jest tu ignorowane, chociaż wyświetlany jest jego argument (na czerwono). Lepszy wynik uzyskamy, wpisując:

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

Jeśli w dokumencie używamy unikodu, a pakietowi `hyperref` podamy opcję `unicode`, to na zakładkach będą mogły wystąpić znaki unikodowe. Poszerza to zakres znaków, które można przekazać w poleceniu `\texorpdfstring`.

4.7.7. Interpretacja pliku źródłowego przez \LaTeX a i $\pdf\LaTeX$ a

W idealnych warunkach dokument powinien się równie dobrze kompilować tak w \LaTeX u, jak i w $\pdf\LaTeX$ u. Problemy w tym względzie mogą wynikać przede wszystkim z włączania grafiki. Prosty sposób jest *systematyczne opuszczanie* rozszerzeń nazw plików w poleceniu `\includegraphics`. Powoduje to wyszukiwanie plików w formacie właściwym dla sposobu kompilacji. Reszta sprowadza się do wygenerowania odpowiednich wersji plików graficznych. \LaTeX poszuka plików `.eps`, podczas gdy $\pdf\LaTeX$ spróbuje włączyć plik o jednym z rozszerzeń `.png`, `.pdf`, `.jpg` lub `.mps` (w podanej kolejności).

Jeśli w $\pdf\LaTeX$ owej wersji dokumentu chcemy użyć innego kodu niż w \LaTeX owej, to do preambuły możemy dodać pakiet `ifpdf`¹⁰. W pakiecie zdefiniowana jest instrukcja `\ifpdf`, która ułatwia pisanie kodu warunkowego. W poniższym przykładzie wersja do druku, w formacie PS, ma być czarno-biała, co obniży koszty w drukarni, podczas gdy wersja PDF, przeznaczona do wyświetlania na ekranie, ma być kolorowa:

```
%& --translate-file=il2-pl
\RequirePackage{ifpdf}
% Sprawdzamy, czy działa pdfTeX:
\ifpdf \documentclass[a4paper,12pt,pdftex]{book}
\else \documentclass[a4paper,12pt,dvips]{book} \fi

\ifpdf \usepackage{lmodern} \fi
% dodajemy hiperłącza
\usepackage[bookmarks,colorlinks,plainpages=false]{hyperref}
\usepackage{polski}
...
```

Pakiet `hyperref` jest tu włączany także w wersji PS. Dzięki temu polecenie `\href` zadziała także w wypadku kompilacji zwykłym \LaTeX em.

Warto odnotować, że w niektórych współczesnych dystrybucjach \TeX a podstawowym programem używanym do kompilacji dokumentów jest właśnie `pdftex`. Przełącza się on na generowanie plików PDF albo DVI, zależnie

¹⁰Pełniejsze wyjaśnienie, kiedy ten pakiet się przydaje, można znaleźć w \TeX owym FAQ-u, w punkcie <http://www.tex.ac.uk/cgi-bin/tefaq2html?label=ifpdf>.

Tabela 2.3: Opcjonalny argument otoczeń `table` i `figure`

Znak	Dopuszczalne miejsce umieszczenia wstawki
<code>h</code>	bez przemieszczenia, dokładnie w miejscu użycia
<code>t</code>	na górze strony
<code>b</code>	na dole strony
<code>p</code>	na stronie zawierającej wyłącznie wstawki
<code>!</code>	ignorując większość parametrów kontrolujących umieszczanie wstawek ^a , przekroczenie wartości, które mogą nie pozwolić na umieszczanie następujących wstawek na stronie.

^aSą to parametry takie jak np. maksymalna dopuszczalna liczba wstawek na stronie

Argument `[!hbp]` oznacza, że tabelę można umieścić w miejscu, w którym pojawia się w pliku źródłowym (`h`), albo na dole strony (`b`), albo wreszcie na osobnej stronie zawierającej wyłącznie wstawki (`p`). Ponadto „`!`” oznacza, że \LaTeX ma pominąć większość parametrów sterujących umieszczaniem wstawek. Jeżeli otoczenia `table` użyto bez opcjonalnego argumentu, to jego domyślnymi wartościami są `[tbp]`.

\LaTeX umieszcza każdą wstawkę zgodnie ze specyfikacją autora podaną w argumentcie *miejsce*. Jeżeli nie może umieścić wstawki na bieżącej stronie, to dołącza ją albo do *kolejki rysunków*, albo do *kolejki tabel*¹². Na początku składania nowej strony \LaTeX sprawdza, czy można ją zapełnić wstawkami czekającymi w kolejce. Jeśli nie jest to możliwe, to pierwsza wstawka każdej z kolejek traktowana jest tak, jak gdyby właśnie pojawiła się w tekście: \LaTeX stara się ją umieścić zgodnie z wartościami parametru *miejsce* (za wyjątkiem `h`, gdyż już nie jest to już oczywiście możliwe). Nowe wstawki dołączane są na koniec odpowiednich kolejek. \LaTeX dba o właściwy porządek wstawek każdego typu. Może się zdarzyć, że pojedynczy rysunek, którego z jakichś względów nie można poprawnie wstawić, „ciągnie” za sobą wszystkie późniejsze rysunki, nawet aż na koniec dokumentu. Dlatego:

Jeżeli \LaTeX nie umieszcza wstawek zgodnie z oczekiwaniami, to z reguły któraś z nich blokuje całą kolejkę, a być może nawet wszystkie kolejki wstawek.

Wyjaśnwszy ów cokolwiek zawily problem umieszczania wstawek, przejdźmy do omówienia kilku pozostałych spraw z nimi związanych. Poleceniem:

```
\caption{tekst}
```

wstawiamy tytuł rysunku lub tabeli. Kolejny numer rysunku bądź tabeli oraz słowo „Rysunek” bądź „Tabela” (lub „Tablica” – zależnie od używanego pakietu polonizacyjnego) zostaną wstawione automatycznie.

¹²Są to kolejki typu FIFO (pierwsze weszło – pierwsze wyjdzie).

`citecolor (=green)` kolor cytowań, czyli odsyłaczy do bibliografii,
`filecolor (=magenta)` kolor odsyłaczy do plików,
`urlcolor (=cyan)` kolor łączy typu URL (adresy poczty elektronicznej bądź sieciowe).

Jeśli zadowolają nas ustawienia domyślne, to wpisujemy:

```
\usepackage[pdftex]{hyperref} %ew. dvips zamiast pdftex
```

Jeśli przeglądarka ma pokazać listę zakładek, a łączy mają być kolorowane (wartości domyślnych `=true` nie trzeba podawać), to:

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

Gdy tworzymy dokumenty PDF przeznaczone do druku czarno-białego, to kolorowanie łączy nie jest najlepszym pomysłem, bo w wydrukach wychodzą one szare, co utrudnia czytanie. Zamiast tego możemy użyć kolorowych ramek, które nie są drukowane:

```
\usepackage{hyperref}
```

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

albo dla łączy używać koloru czarnego:

```
\usepackage{hyperref}
```

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

Jeśli chcemy dodać informacji o dokumencie PDF, które Acrobat wyświetla w oknie *Document Properties*:

```
\usepackage[pdftauthor={P. Desproges},pdftitle={Des femmes
qui tombent},pdftex]{hyperref}
```

Oprócz hiperłączy tworzonych automatycznie możemy umieszczać własne:

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

Przykładowo, w wyniku wykonania następującego fragmentu:

```
Na stronie \href{http://www.ctan.org}{CTAN}.
```

zostanie wydrukowane słowo „CTAN”, kliknięcie w to słowo przeniesie nas na internetową stronę CTAN-u.

Jeśli docelowe miejsce łączy nie jest URL-em, lecz plikiem lokalnym, to możemy użyć polecenia `\href` w następującej postaci:

```
Pełny dokument jest w~\href{manual.pdf}{pliku}.
```

Generuje to tekst „Pełny dokument jest w [pliku](#)”. Klikając w słowo „[pliku](#)”, otworzymy plik `manual.pdf`. (Nazwa pliku jest interpretowana względem położenia dokumentu PDF).

Autorka dokumentu może zachęcić czytelników do korespondencji elektronicznej, wpisując instrukcję `\href` jako fragment polecenia `\author` na stronie tytułowej dokumentu:

2.13. Ochrona poleceń kruchych

Niektórych poleceń nie można umieszczać wewnątrz argumentów innych poleceń, na przykład polecenie `\footnote` albo `\verb` nie może się pojawić w argumencie polecenia `\section` czy `\caption`. Kompilacja dokumentu zakończy się w takiej sytuacji błędem, a takie polecenia nazywamy *kruchymi* (ang. *fragile command*).

Polecenia kruche wymagają „ochrony”. Możemy je chronić, umieszczając przed nimi polecenie `\protect`. Polecenie `\protect` odnosi się wyłącznie do instrukcji znajdującej się tuż za nim, nie obejmuje swym działaniem nawet argumentów tej instrukcji. W większości wypadków nadmiarowe użycie `\protect` nie powoduje szkód.

```
\section{Jestem przezorny%
```

```
\protect\footnote{i~chronię przypisy}}
```

Uwaga: wiele osób lubi dodawać przypisy do śródtytułów. Naszym zdaniem jest to zły i komplikujący życie zwyczaj; ostatecznie zawsze można umieścić przypis w pierwszym akapicie pod śródtytułem. Rób jak uważasz, ale czy wiesz, jak usunąć numer przypisu ze spisu treści?

2.14. Listy

Do pisania listów można użyć klasy `letter`. Struktura pliku źródłowego tej klasy różni się od dokumentów z klasy `article` czy `book`. Klasę `letter` zaprojektowano tak, by bezproblemowo dało się napisać zarówno pojedynczy list do przysłówiowej „cioci”, jak też setki listów do różnych osób (korespondencja seryjna).

Jeśli imię, nazwisko i adres nadawcy mają być identyczne we wszystkich listach, to deklarujemy je poleceniem:

```
\address{imię\\nazwisko\\ adres...}
```

Użycie instrukcji `\\` w adresie powoduje złamanie wiersza.

Polecenie `\signature` służy do zadeklarowania podpisu pod listem. Wewnątrz argumentu tego polecenia instrukcja `\\` służy do rozpoczęcia nowego wiersza, przykładowo:

```
Dyr. E.~K.~Tor,\\ Przewodniczący
Zastępcy
```

```
Dyr. E. K. Tor,
Przewodniczący Zastępcy
```

Do nagłówka listu automatycznie wstawiana jest bieżąca data. Aby wstawić inną, należy zastosować deklarację `\date`:

```
\date{16 Czerwca 1963~r.}
```

L^AT_EX umie korzystać zarówno z fontów PK, jak i z fontów Type 1, zaś pdfL^AT_EX potrafi dodatkowo używać fontów TrueType i OpenType. Jednakże dokument z tradycyjnymi fontami PK bardzo źle wygląda w przeglądarce Acrobat Reader. Najlepiej używać wyłącznie fontów Type 1, bo wówczas dokumenty zawsze wyświetlają się poprawnie. Współczesne instalacje T_EXa są domyślnie skonfigurowane tak, by używane były fonty Type 1⁹. W zależności od sposobu polonizacji dokumentu (por. punkt 2.5.1) L^AT_EX albo wykorzysta obwiedniowe wersje fontów PL, albo przełączy się na fonty z pakietu cm-super, które są implementacją fontów EC w technologii Type 1.

Ostatnio dostępna stała się wysokiej jakości rodzina fontów obwiedniowych LM (*Latin Modern*). Jeśli dysponujemy świeżej daty instalacją T_EXa, to jest całkiem możliwe, że zawiera ona już tę rodzinę. Wystarczy w takim wypadku dodać do preambuły:

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

Uwaga! Rodzina LM nie zawiera na razie fontów matematycznych, dlatego do składu formuł i tak potrzebne są albo fonty PL, albo oryginalne fonty L^AT_EXowe.

Uwaga! Jeśli w pliku o rozszerzeniu .log, zawierającym raport z kompilacji, pojawią się komunikaty w rodzaju:

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

to oznacza, że któryś z fontów potrzebnych do przetworzenia dokumentu nie został odnaleziony. Problem taki wymaga rzetelnej naprawy, gdyż przeglądarka plików PDF *w ogóle nie wyświetli stron ze znakami z brakującego fontu*.

4.7.3. Dołączanie grafiki

Do włączania grafiki najlepiej korzystać z pakietu `graphicx` (zobacz str. 65). Po wpisaniu `pdftex` jako opcji *sterownik* pakiet zadziała też w pdfL^AT_EXu:

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

W tym przykładzie pojawia się również pakiet `color`, jako że stosowanie kolorów w sieci jest naturalne.

Tyle dobrych wieści. Złe są takie, że grafika w formacie EPS nie działa w pdfL^AT_EXu. Jeśli w argumencie instrukcji `\includegraphics` pominiemy rozszerzenie nazwy pliku, to pakiet `graphicx` będzie szukał pliku w formacie zależnym od opcji *sterownik*. W programie pdfT_EX dopuszczalny jest jeden z formatów: `.png`, `.pdf`, `.jpg`, `.mps` (METAPOST), ale *nie* `.eps`.

⁹Aby się przekonać, czy tak jest w wypadku twojej instalacji, utwórz plik PDF (na przykład za pomocą pdfL^AT_EX) i wyświetl go w oknie przeglądarki Acrobat Reader. Jeżeli fonty na ekranie są nieczytelne i mają poszarpane brzegi, to masz problem. W nowych dystrybucjach L^AT_EXa problem ten nie powinien wystąpić, więc jeżeli go napotkalesz, to przypuszczalnie masz zainstalowaną zbyt okrojona wersję systemu T_EX.

Rozdział 3

Wyrażenia matematyczne

Nareszcie! W tym rozdziale poznasz najlepszą stronę T_EXa, czyli skład wzorów matematycznych. Ostrzegamy jednak, że przedstawimy tu jedynie absolutne podstawy. Chociaż wystarczają one większości użytkowników, to nie załamuj rąk, jeśli nie poradzisz sobie z jakimś skomplikowanym wzorem, lecz zapoznaj się z możliwościami $\mathcal{A}\mathcal{M}\mathcal{S}$ -L^AT_EXa lub innego wyspecjalizowanego pakietu¹.

3.1. Wstęp

Do składu wyrażeń matematycznych mamy w L^AT_EXu specjalny *tryb matematyczny*. Wzory wpisuje się między znakami \$ i \$, między parami znaków \ (i \) albo między `\begin{math}` oraz `\end{math}`.

`a` do kwadratu plus `~b`
do kwadratu równa się `~c`
do kwadratu. Albo, stosując
bardziej matematyczne
podejście: `$c^{2}=a^{2}+b^{2}$`.

a do kwadratu plus *b* do kwadratu równa się *c* do kwadratu. Albo, stosując bardziej matematyczne podejście: $c^2 = a^2 + b^2$.

`{\TeX}` należy wymawiać jako
`$\tau\epsilon\chi$`. `\\[6pt]`
`100~m^{3}$` wody. `\\[6pt]`
To płynie z `~mojego~$heartsuit$`.

T_EX należy wymawiać jako $\tau\epsilon\chi$.
100 m³ wody.
To płynie z mojego ♡.

Składając większe wzory, powinniśmy je eksponować, to znaczy wstawiać między akapitami, w osobnym wierszu. Takie wzory umieszcza się albo między parami znaków \ [i \], albo wewnątrz otoczenia `displaymath`. Ta ostatnia konstrukcja dotyczy tworzenia wzorów bez numeracji:

¹Pod egidą Amerykańskiego Towarzystwa Matematycznego (*American Mathematical Society*) powstało istotne rozszerzenie L^AT_EXa. Wiele przykładów w tym rozdziale korzysta z tego rozszerzenia, które jest dołączane do wszystkich współczesnych dystrybucji T_EXa. Jeśli w twojej go nie ma, to znajdziesz je pod adresem `CTAN://macros/latex/packages/amslatex`.

4.7. Tworzenie plików PDF: pdfL^AT_EX i hyperref

PDF (*Portable Document Format*) to format hipertekstowy. Podobnie jak na stronie internetowej, niektóre słowa są w dokumencie zaznaczone jako hiperłącza. Łączą one miejsca w dokumencie z innymi jego miejscami bądź nawet z innymi dokumentami; gdy klikamy w takie hiperłącze, jesteśmy przeniesieni do miejsca docelowego. W kontekście L^AT_EXa oznacza to, że wszystkie wystąpienia poleceń `\ref` i `\pageref` stają się hiperłączami. Na dodatek spis treści, skorowidz i wszystkie podobne struktury stają się zestawami łączy.

Większość spotykanych dziś w Internecie stron jest napisana w języku HTML (*HyperText Markup Language*). Kiedy przychodzi do pisania pracy naukowej, format ten ma dwie wady:

1. Nie ma dobrej metody włączania wzorów matematycznych. Chociaż został zdefiniowany odpowiedni standard, to jednak większość przeglądarek go nie obsługuje, brakuje też w komputerach odpowiednich fontów.
2. Dokumenty HTML można co prawda drukować, ale wydruki wyglądają różnie w różnych przeglądarkach, a także na różnych platformach. Od jakości, do której przyzwyczajają nas L^AT_EX, wydruki te dzieli przepaść.

Próbowano już na wiele sposobów napisać program tłumaczący język L^AT_EXa na HTML. Niektóre z tych programów dają nawet niezłe wyniki – w tym sensie, że są w stanie ze źródła w standardowym L^AT_EXu wygenerować czytelny dokument HTML. Wszystkim jednak zawsze czegoś brakuje. Gdy tylko zachodzi konieczność użycia bardziej wyrafinowanych możliwości L^AT_EXa bądź jego pakietów zewnętrznych, jakość spada. Autorom, którym zależy na publikowaniu w sieci WWW, a jednocześnie chcą utrzymać wysoką jakość typograficzną, pozostaje w zasadzie wyłącznie PDF. Większość współczesnych przeglądarek sieciowych potrafi wyświetlać dokumenty w tym formacie.

Chociaż przeglądarki do plików w formatach DVI i PS są dostępne na prawie każdej platformie, to jeszcze powszechniej spotyka się przeglądarki do plików PDF, takie jak Acrobat Reader i xpdf. Tak więc, jeśli naszą pracę udostępnimy w postaci PDF, to stanie się ona łatwiej dostępna dla grona potencjalnych czytelników.

4.7.1. Tworzenie dokumentów PDF w L^AT_EXu

Są dwa sposoby⁷ zamiany plików źródłowych L^AT_EXa na dokument PDF. Pierwszy z nich polega na zamianie pliku DVI na dokument PS (zwykle za pomocą programu `dvips`), który następnie jest transformowany do formatu PDF bądź programem `ghostscript`, bądź z użyciem komercyjnego narzędzia Acrobat Distiller, firmy Adobe. Drugi sposób to bezpośrednie generowanie

⁷Tych sposobów jest więcej, ale przedstawione dwa są zdecydowanie najczęściej używane.

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

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

Matematycy potrafią być niezwykle staranni w doborze symboli. Na przykład we wzorach, w których występują oznaczenia zbiorów (jak powyższy), często stosuje się krój, w którym te oznaczenia przypominają odmianę „grubą”, pisaną kredą na tablicy ($\mathbb{A}, \mathbb{B}, \mathbb{C} \dots$). Symbole takie wstawiamy do wzoru poleceniem `\mathbb{b}` z pakietu `amsmath` lub `amssymb`. Ostatni przykład wygląda wtedy następująco:

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

$$x^2 \geq 0 \quad \text{dla każdego } x \in \mathbb{R}$$

3.2. Grupowanie

Argumentem większości instrukcji do składu matematyki jest tylko jeden znak – ten, który następuje tuż po instrukcji. Jeżeli polecenie ma dotyczyć grupy znaków, to należy je umieścić wewnątrz pary nawiasów klamrowych `{...}`:

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

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

3.3. Części składowe wyrażeń matematycznych

W tym podrozdziale opiszemy ważniejsze instrukcje do składu wyrażeń. Zestawienie wszystkich dostępnych symboli i znaków znajduje się w punkcie 3.10 na stronie 58.

Małe litery alfabetu greckiego wprowadzamy, używając instrukcji typu: `\alpha`, `\beta`, `\gamma` itd., a **duże**²: `\Delta`, `\Gamma` itd.:

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

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

Indeksy górne i wykładniki otrzymujemy za pomocą znaku `^`, a **dolne** – stosując podkreślenie `_`:

²Obecnie brakuje dużej litery *Alpha* i wygląda ona identycznie jak pierwsza litera *A* alfabetu łacińskiego. Sytuacja ta ma się zmienić po wprowadzeniu nowego sposobu kodowania symboli matematycznych.

Za pomocą \LaTeX a można sobie łatwo poradzić z automatycznym umieszczeniem odpowiedniej informacji w żywej paginie. Rozwiązanie jest następujące. W definicjach poleceń składających paginę wykonanie instrukcji `\rightmark` oraz `\leftmark` wstawia odpowiedni tytuł rozdziału, punktu lub cokolwiek innego. Poleceniom `\rightmark` i `\leftmark` jest nadawane nowe znaczenie (nowe wartości) za każdym wykonaniem instrukcji składania tytułu rozdziału i punktu (`\chapter`, `\section`).

W rzeczywistości, `\chapter` oraz inne polecenia podziału logicznego dokumentu nie zmieniają znaczenia poleceń `\rightmark` i `\leftmark`. Odwołują się one natomiast do poleceń `\chaptermark`, `\sectionmark` oraz `\subsectionmark`. Dopiero użycie tych instrukcji powoduje zmianę definicji poleceń `\rightmark` i `\leftmark`.

Do zmiany postaci tytułu rozdziału w paginie wystarczy modyfikacja polecenia `\chaptermark`. Rysunek 4.1 przedstawia takie wykorzystanie pakietu, że paginy górne będą wyglądać mniej więcej tak jak w tym podręczniku. Kompletny opis pakietu znajduje się w jego dokumentacji.

4.5. Pakiet verbatim

Pakiet `verbatim` udostępnia poprawioną wersję standardowego otoczenia `verbatim` (opisanego na stronie 38). Oprócz wielu drobnych, chociaż istotnych ulepszeń pakiet udostępnia polecenie:

```
\verbatiminput{plik}
```

które dołącza do dokumentu plik tekstowy tak, jakby jego zawartość znajdowała się wewnątrz otoczenia `verbatim`.

Pakiet `verbatim` jest częścią grupy pakietów o nazwie „tools”. Wchodzą one w skład standardowej dystrybucji \LaTeX a. Więcej szczegółów znajdziemy w [25].

4.6. Instalowanie dodatkowych pakietów

Ten punkt jest adresowany raczej do tych czytelników, którzy mają już doświadczenie w \LaTeX u, do składania dokumentów używają własnych poleceń i otoczeń, a odczuwają potrzebę uporządkowania sobie środowiska pracy bądź udostępnienia go innym. Czytelnicy początkujący mogą bez wielkiej straty pominąć ten punkt i kontynuować czytanie od następnego punktu.

W większości instalacji \LaTeX a wbudowany jest bogaty zestaw pakietów dodatkowych, jeszcze ich więcej można znaleźć w sieci. Głównym miejscem, gdzie można je znaleźć, jest archiwum CTAN (<http://www.ctan.org/>).

Pakiety takie jak `geometry` albo `hyphenat`, a także wiele innych, składają się na ogół z dwóch plików: jednego o rozszerzeniu `.ins` i drugiego

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

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

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

W Polsce nazwy niektórych funkcji trygonometrycznych różnią się od anglosaskich. Pakiet `polski` – po dołączeniu do dokumentu – zmienia na życzenie standardowe funkcje \LaTeX a na zgodne ze zwyczajami polskimi. Oto angielskie oryginały: `tan`, `coth`, `tanh`, `arccos`, `arcsin` i ich polskie odpowiedniki: `tg`, `ctgh`, `tgh`, `arc cos`, `arc sin`.

Dla funkcji typu modulo istnieją dwie instrukcje: `\bmod` dla binarnego operatora „ $a \bmod b$ ” oraz `\pmod` do składu takich wyrażeń jak „ $x \equiv a \pmod{b}$ ”.

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

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

Ułamki piętrowe składa się poleceniem `\frac{...}{...}`. Do ułamków zwykłych czasami lepiej stosować kreskę ukośną, zwłaszcza w wypadku niewielkich porcji materiału ułamkowego, jak „ $1/2$ ”:

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

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

Do składu dwumianów lub podobnych konstrukcji możemy użyć polecenia `{... \choose ...}` albo `{... \atop ...}`. Instrukcja `\atop` daje efekt taki jak `\choose`, tyle że bez nawiasów:

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

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

W pakiecie `amsmath` do składu dwumianu Newtona dostępne jest polecenie `\binom`:

```
\begin{displaymath}
\binom{n}{k} \quad \mathrm{C}_n^k
\end{displaymath}
```

$$\binom{n}{k} \quad C_n^k$$

Hasło wstawiamy do skorowidza poleceniem:

```
\index{hasło}
```

gdzie *hasło* oznacza pozycję w skorowidzu. Polecenie `\index{hasło}` umieszczamy w pliku źródłowym bezpośrednio w miejscu związanym z określonym hasłem. W tabeli 4.2 przedstawiono przykłady użycia *hasel*.

Tabela 4.2: Przykłady składni polecenia `\index`

Przykład	Hasło	Uwagi
<code>\index{kot}</code>	kot, 1	hasło pierwszego stopnia
<code>\index{kot!rudy}</code>	rudy, 3	hasło drugiego stopnia
<code>\index{Sam@textsl{Sam}}</code>	Sam, 2	hasło sformatowane
<code>\index{Lin@textbf{Lin}}</code>	Lin, 7	hasło sformatowane
<code>\index{Cadiz@C'adiz}</code>	Cádiz, 77	poprawne sortowanie
<code>\index{Jenny textbf}</code>	Jenny, 3	formatowanie numeru strony
<code>\index{Joe textit}</code>	Joe, 5	formatowanie numeru strony
<code>\index{kot see{felis}}</code>	kot, zob. felis	relacja pomiędzy hasłami

Aby wstawić do skorowidza literalnie znak `!`, `@` albo `|` należy poprzedzić go znakiem `"`.

Podczas przetwarzania pliku źródłowego przez \LaTeX a każda instrukcja `\index` powoduje, że do pliku pomocniczego zostaje zapisana odpowiednia pozycja skorowidza wraz z bieżącym numerem strony. Plik pomocniczy nosi tę samą nazwę co główny plik źródłowy, ma jednak rozszerzenie `.idx`. Trzeba go następnie przetworzyć programem `makeindex`, w taki oto sposób:

```
makeindex plik
```

W rezultacie program `makeindex` tworzy posortowany skorowidz i zapisuje go do pliku o nazwie identycznej z nazwą głównego pliku źródłowego i o rozszerzeniu `.ind`. Jeżeli jeszcze raz przetworzymy plik źródłowy, to tym razem skorowidz zostanie włączony w miejscu wystąpienia polecenia:

```
\printindex
```

Pakiet `showidx`, wchodzący w skład standardowej dystrybucji \LaTeX 2 ϵ , drukuje pozycje skorowidza na lewym marginesie szpalty. Jest on dość przydatny do sprawdzania i korygowania skorowidza.

Reguły sortowania skorowidza zależą oczywiście od języka, w którym pisany dokument. Pod tym względem program `makeindex` jest nieprzydatny dla dokumentów polskojęzycznych, gdyż sortuje hasła jedynie według zasad języka angielskiego. Wersją programu `makeindex` przystosowaną do polskich reguł sortowania wyrazów jest `plmindex` (autor: Włodzimierz Macewicz).

`\left` oraz ogranicznika wymaga nawiasu zamykającego poprzedzonego poleceniem `\right`. Gdy ogranicznik ma się pojawić tylko po jednej stronie, wówczas po drugiej należy użyć konstrukcji z kropką: `\left.` po lewej albo `\right.` po prawej:

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

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

W pewnych sytuacjach trzeba samemu określić właściwą wielkość ogranicznika. Do tego celu służą instrukcje `\big`, `\Big`, `\bigg` oraz `\Bigg`, poprzedzające odpowiedni ogranicznik⁵:

```

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

```

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

Wielokropki w wyrażeniach matematycznych wprowadzamy poleceniem `\ldots`. Kropki pojawiają się wtedy na linii podstawowej, to znaczy na jednakowej wysokości z przecinkiem czy kropką. Instrukcja `\cdots` wstawia natomiast inny rodzaj wielokropka, w którym kropki znajdują się w osi znaków `+`, `-`, `=`:

```
\begin{displaymath}
x_1, \ldots, x_n \quad x_1 + \cdots + x_n
\end{displaymath}
```

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

Są jeszcze instrukcje `\vdots` oraz `\ddots`. Pierwsza z nich generuje wielokropki pionowy, a druga – skośny (zobacz przykład w punkcie 3.5).

3.4. Odstępy w trybie matematycznym

Zdarzają się sytuacje, kiedy wielkość odstępów wewnątrz wyrażeń matematycznych jest niepoprawna. Można je skorygować odpowiednimi instrukcjami. Do wprowadzania niewielkich odstępów służy kilka poleceń: `\`, wstawia odstęp równy $\frac{3}{18}$ em⁶ (μ), `\:` pozwala uzyskać odstęp równy $\frac{4}{18}$ em (μ) a `\;` – odstęp równy $\frac{5}{18}$ em (μ). Użycie instrukcji `_` (spacja po znaku `\`) prowadzi

⁵Polecenia te nie działają zgodnie z oczekiwaniami, gdy uprzednio zmieniono stopień pisma, na przykład użyto opcji `11pt` lub `12pt`. W takiej sytuacji należy skorzystać z pakietu `exscale` albo pakietu `amsmath`.

⁶W programach komputerowych przyjęło się stosowanie jednostki *em* równej szerokości litery „M” w bieżącym kroju pisma. Por. też [20].

(`\textwidth` w języku \LaTeX), między rysunkami wstawiony zostanie odstęp wielkości zwykłej spacji, co wynika z zasady \LaTeX a, że pojedyncza zmiana wiersza jest traktowana jak odstęp. Całość zostanie wyśrodkowana w osi szpalty. Pamiętajmy, żeby do tego stosować polecenie `\centering` a nie otoczenie `center`, gdyż to drugie wstawia dodatkowy a zbędny odstęp pionowy.

Aby efekt końcowy na wydruku był zadowalający, rysunki powinny mieć identyczne wymiary. Inaczej, albo poszczególne obrazki będą optycznie niezgodnie (np. sowa uszata będzie 3 razy większa od puchacza), albo też poszczególne rysunki składowe będą miały różne wymiary (rysunek z sową uszatą będzie dwa razy mniejszy niż rysunek z puchaczem).

Czasami rysunki czy tabele zmieściłyby się na stronie, gdyby obrócić je o 90° . W tym celu można skorzystać z pakietu `rotating`. Do obracania dowolnego fragmentu tekstu, pudełka (por. punkt 6.6), tabeli itp. należy używać otoczenia `rotate`:

```
\begin{rotate}{kąt} ... \end{rotate}
```

Do obrócenia obiektu o kąt 90° należy użyć otoczenia `sideways`:

```
\begin{sideways} ... \end{sideways}
```

Do obrócenia tabeli łącznie z podpisem pod tabelą należy zastosować otoczenie `sidewaystable` (uwaga: tabela jest umieszczana na osobnej stronie):

```
\begin{sidewaystable} ... \end{sidewaystable}
```

Do obrócenia rysunku łącznie z podpisem pod rysunkiem należy użyć otoczenia `sidewaysfigure` (uwaga: rysunek jest umieszczany na osobnej stronie):

```
\begin{sidewaysfigure} ... \end{sidewaysfigure}
```

4.2. Spis literatury

Do przygotowania spisu literatury służy otoczenie `thebibliography`. Każda pozycja w tym otoczeniu ma postać polecenia:

```
\bibitem{etykieta}
```

Etykiety można następnie użyć do zacytowania w dokumencie tej pozycji, czyli książki, artykułu bądź pracy konferencyjnej:

```
\cite{etykieta}
```

Numerowanie pozycji literaturowych jest automatyczne (polecenia `\bibitem` i `\cite` działają podobnie jak opisane już instrukcje `\label` i `\ref`). Otoczenie `thebibliography` ma jeden parametr, który powinien zawierać tekst przynajmniej tak szeroki jak najszersza etykieta ze spisu. W poniższym przykładzie zapis `99` oznacza, że numery pozycji w spisie będą co najwyżej dwucyfrowe.

Otoczeniem `array` możemy się posłużyć także do składania wyrażeń zawierających tylko jeden ogranicznik, po prawej lub po lewej stronie, stosując konstrukcję z kropką `\right.` lub `\left.`:

```
\begin{displaymath}
y = \left\{ \begin{array}{l} a \\ b+x \\ l \end{array} \right. \begin{array}{l} \text{jeżeli } d > c \\ \text{rano} \\ \text{w ciągu dnia} \end{array} \\
\end{displaymath}
```

$$y = \begin{cases} a & \text{jeżeli } d > c \\ b+x & \text{rano} \\ l & \text{w ciągu dnia} \end{cases}$$

Podobnie jak w wypadku otoczenia `tabular`, także w otoczeniu `array` można wstawiać pionowe i poziome kreski, np. oddzielające poszczególne rubryki macierzy:

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

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

Do składu wyrażeń wielowierszowych można użyć – zamiast otoczenia `equation` – otoczeń `eqnarray` lub `eqnarray*`. W otoczeniu `eqnarray` każdy wiersz zawartego w nim wyrażenia otrzymuje osobny numer; w otoczeniu `eqnarray*` wiersze nie są numerowane. Otoczenia `eqnarray` oraz `eqnarray*` działają jak trójkolumnowa tabela w układzie `{rc1}`. W takiej tabeli w środkowej kolumnie wstawiamy zwykle znaki równości lub nierówności. Poleceniem `\\` łamiemy zawartość otoczenia na wiersze:

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

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

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

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

Zwróćmy uwagę, że odstęp po obu stronach znaku równości jest dość duży. Można go zmniejszyć poleceniem `\setlength\arraycolsep{2pt}`, które najlepiej jest umieścić w preambule dokumentu.

\LaTeX nie dzieli automatycznie długich wyrażeń, niemieszczących się w jednym wierszu. Musimy to zrobić sami. Najczęściej stosuje się takie sposoby:

1. W programie graficznym zachowujemy rysunek w formacie EPS³ lub konwertujemy rysunek na format EPS, jeżeli dysponujemy już gotową grafiką ale w innym formacie⁴.
2. Dołączamy pakiet `graphicx` do preambuły dokumentu:

```
\usepackage[dvi-ps]{graphicx}
```

gdzie `dvi-ps` oznacza nazwę programu do konwersji pliku wyjściowego `.dvi` na plik postscriptowy. Najczęściej używanym do tego celu programem jest `dvips`. Nazwa sterownika jest tu konieczna, gdyż brakuje standardu dotyczącego dołączania grafiki postscriptowej w dokumentach \TeX owych. Kierując się nazwą sterownika, pakiet `graphicx` potrafi do wynikowego pliku `.dvi` włączyć informację potrzebną do tego, by interpreter Postscriptu (w drukarce bądź w komputerze) poprawnie umieścił rysunek na wydruku.

3. Instrukcją:

```
\includegraphics[klucz=wartość,...]{plik}
```

włączamy *plik* do dokumentu. Parametr opcjonalny jest listą oddzielonych przecinkami *kluczy*, o określonych przez nas *wartościach*. Klucze wykorzystujemy do zmiany parametrów dołączanego rysunku, takich jak szerokość, wysokość czy kąt obrotu. W tabeli 4.1 zamieszczono najważniejsze klucze.

Tabela 4.1: Znaczenie ważniejszych kluczy polecenia `\includegraphics`

<code>width=w</code>	skalowanie rysunku do podanej szerokości <i>w</i>
<code>height=h</code>	skalowanie rysunku do podanej wysokości <i>h</i>
<code>angle=a</code>	obrót o kąt <i>a</i> (przeciwnie do ruchu wskazówek zegara)
<code>scale=s</code>	równomierne przeskalowanie w skali <i>s</i>

Poniższy przykład pomoże zrozumieć całą ideę:

³Jeżeli używany przez nas program graficzny na to *nie pozwala*, to spróbujmy zainstalować sterownik do drukarki postscriptowej, jak Apple Laser Writer, i drukować tym sterownikiem do pliku. Przy odrobinie szczęścia otrzymamy dokument w formacie EPS. Pamiętajmy, że plik EPS może zawierać tylko jedną stronę. Sterowniki niektórych drukarek można jawnie ustawić do tworzenia plików wyjściowych właśnie w tym formacie.

⁴W wypadku grafiki obwiedniowej i rastrowej, można do tego celu użyć programu ImageMagick (<http://www.imagemagick.org/>). Dobrym programem do konwersji, ale tylko grafiki rastrowej, jest gimp (<http://www.gimp.org>). Do konwersji zdjęć albo innych grafik w formacie JPG wygodnym narzędziem jest `jpeg2ps` <http://gnuwin32.sourceforge.net/packages/jpeg2ps.htm>. Niestety konwersja plików zapisanych w promowanym przez firmę Microsoft formacie WMF/EMF (MS Visio, MS Office, itp.) często daje opłakane rezultaty, ale to już jest wina wspomnianej firmy i stosowanej przez nią „strategii biznesowej”...

3.7. Stopień pisma

W trybie matematycznym stopień pisma dobierany jest automatycznie, zależnie od kontekstu. Indeksy górne \LaTeX składa na przykład mniejszą czcionką. Gdy wewnątrz wyrażenia matematycznego zachodzi potrzeba złożenia fragmentu normalnego tekstu, a użyjemy polecenia `\textrm`, to nie zadziała mechanizm przełączania stopni pisma. Będzie tak, ponieważ polecenie `\textrm` powoduje tymczasowe przejście do trybu tekstowego.

Zamiast `\textrm` można użyć polecenia `\mathrm`, które zachowuje mechanizm zmiany stopnia pisma. Pamiętajmy jednak, że działa ono poprawnie w zasadzie tylko dla pojedynczych wyrazów, ponieważ znaki odstępów są ignorowane. Ponadto nie działa zgodnie z oczekiwaniami mechanizm akcentowania⁹:

```
\begin{equation}
2^{\textrm{nd rd}}^{\textrm{th}} \quad \quad \quad 2^{\mathrm{nd rd}}^{\mathrm{th}} \quad (3.10)
\end{equation}
```

Czasami musimy wyraźnie określić stopień pisma, jakim chcemy się posłużyć. W trybie matematycznym możemy do tego stosować cztery następujące polecenia:

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

Zmiana stylu dotyczy także sposobu składania indeksów górnych i dolnych, jak granice sumowania czy całkowania:

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

Powyższy przykład ilustruje sytuację, w której należy zastosować polecenie `\biggl` bądź `\biggr`, ponieważ nawiasy pochodzące z konstrukcji `\left[` oraz `\right]` byłyby zbyt małe.

⁹W pakiecie $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\LaTeX$ polecenie `\textrm` zostało poprawione i umożliwia automatyczną zmianę stopnia pisma zależnie od kontekstu. Pakiet ten nie tylko poprawia działanie `\textrm`, ale też definiuje instrukcję `\text` do wstawiania fragmentów „normalnego” tekstu wewnątrz wzorów.

Tabela 3.17: Różne symbole (pakiet AMS)

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

Tabela 3.18: Kroje pisma dostępne w trybie matematycznym

Przykład	Polecenie	Wymagany pakiet
$ABCdef$	<code>\mathrm{ABCdef}</code>	
$ABCdef$	<code>\mathit{ABCdef}</code>	
$ABCdef$	<code>\mathnormal{ABCdef}</code>	
ABC	<code>\mathcal{ABC}</code>	
ABC	<code>\mathcal{ABC}</code>	eucal z opcją <code>mathcal</code> lub
ABC	<code>\mathscr{ABC}</code>	eucal z opcją <code>mathscr</code>
$\frac{ABC}{def}$	<code>\mathfrak{ABCdef}</code>	eufrak
\mathbb{ABC}	<code>\mathbb{ABC}</code>	amssymb lub <code>amssymb</code>

```
\newtheorem{mur}{Murphy}[section]
\begin{mur} Jeżeli coś można
wykonać na dwa lub więcej sposobów,
przy czym jeden z nich prowadzi do
katastrofy, to sposób ten zostanie
przez kogoś wybrany.\end{mur}
```

Murphy 3.8.1. *Jeżeli coś można wykonać na dwa lub więcej sposobów, przy czym jeden z nich prowadzi do katastrofy, to sposób ten zostanie przez kogoś wybrany.*

Numeracja twierdzenia „Murphy’ego” jest tu powiązana z numeracją kolejnych punktów. Można też do numerowania twierdzeń stosować inne jednostki podziału dokumentu, jak rozdziały czy podpunkty.

W pakiecie `amsthm` znajduje się też otoczenie `proof` do zapisywania dowodów:

```
\begin{proof}
Banalne. Użyj \[E=mc^2\]
\end{proof}
```

Dowód. Banalne. Użyj
 $E = mc^2$

□

Polecenie `\qedhere` pozwala wstawić symbol „końca dowodu” w określonym miejscu zamiast domyślnego umieszczenia go w oddzielnym wierszu:

```
\begin{proof}
Banalne. Użyj \[E=mc^2 \qedhere\]
\end{proof}
```

Dowód. Banalne. Użyj
 $E = mc^2$

□

3.9. Symbole półgrube

Wstawianie symboli półgrubych jest w $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ u zadaniem dość trudnym. Być może jest tak celowo, ponieważ składacze-amatorzy mają skłonność do ich nadużywania. Poleceniem `\mathbf` uzyskamy odmianę półgrubą. Nie będzie to niestety półgruba kursywa, jaką zwykle składane są symbole matematyczne. Istnieje co prawda polecenie `\boldmath`, ale można go użyć jedynie *poza trybem matematycznym*. Jego działanie obejmuje również symbole.

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

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

Zauważmy, że przecinek też został złożony w odmianie półgrubej, co z reguły jest niepożądanym efektem.

Pakiet `amssymb`, dołączany przez `amsmath`, czyni zadanie dużo łatwiejszym. W pakiecie tym dostępne są polecenia `\boldsymbol` oraz `\pmb`. Instrukcja `\pmb` imituje znak półgruby przez wydrukowanie dwóch nakładających się na siebie znaków, złożonych w odmianie normalnej. Można tym

Tabela 3.13: Symbole relacji (pakiet AMS)

\lessdot	\gtrdot	\doteqdot lub \Doteq
\leqslant	\geqslant	\risingdotseq
\leqslantless	\leqslantgtr	\fallingdotseq
\leqq	\geqq	\eqcirc
\lll lub \llless	\ggg lub \gggtr	\circeq
\lesssim	\gtrsim	\triangleq
\lessapprox	\gtrapprox	\bumpeq
\lessgtr	\gtrless	\Bumpeq
\lesseqgtr	\gtreqless	\thicksim
\lesseqqgtr	\gtreqqless	\thickapprox
\preccurlyeq	\succcurlyeq	\approxeq
\curlyeqprec	\curlyeqsucc	\backsim
\prec	\succ	\backsimeq
\precapprox	\succapprox	\vDash
\subseteq	\supseteq	\Vdash
\Subset	\Supset	\Vvdash
\sqsubset	\sqsupset	\backepsilon
\therefore	\because	\varpropto
\shortmid	\shortparallel	\between
\smallsmile	\smallfrown	\pitchfork
\vartriangleleft	\vartriangleright	\blacktriangleleft
\trianglelefteq	\trianglerighteq	\blacktriangleright

Tabela 3.14: Strzałki (pakiet AMS)

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

Tabela 3.3: Symbole relacji

Odpowiednie symbole negacji można utworzyć, poprzedzając każde z poniższych poleceń instrukcją `\not`.

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

^a Dostępne po dołączeniu pakietu `latexsym`.

przed i za znakiem, bo traktuje dwukropek jako znak relacji. Oto ilustracja tej różnicy:

Nie $f:A\to B$,
lecz $f:\text{colon } A\to B$

Nie $f : A \rightarrow B$, lecz $f : A \to B$

Jeżeli przecinek oddziela część całkowitą liczby od części dziesiętnej, to wskazane jest zakazać `LATEX`-owi wstawiania dodatkowego odstępu, co zwykle robi w trybie matematycznym. Wystarczy w tym celu otoczyć przecinek parą nawiasów klamrowych. Porównajmy:

Zamiast $\$22,115\$$ lepiej $\$22\{,\}115\$$

Zamiast 22, 115 lepiej 22,115

Znaki mniejszy-lub-równy i większy-lub-równy mają kształt różny od stosowanego w krajach anglosaskich. Po dołączeniu pakietu polski standardowe w `LATEX`u polecenia `\leq` oraz `\geq` generują polskie wersje tych relacji, to znaczy \leq i \geq zamiast \leq i \geq .