

Не очень краткое введение в L^AT_EX 2_ε

или E_T_X 2_ε за 129 минут

Tobias Oetiker

Hubert Partl, Irene Hyna и Elisabeth Schlegl

Версия 4.12, 13 April, 2003

Перевод: Б. Тоботрас, 22 мая 2003 г.

Copyright ©1995-2002 Tobias Oetiker and all the Contributors to LShort. All rights reserved.

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

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

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

Благодарности

Многое из включенного в это введение материала исходит из австрийского введения в $\text{\LaTeX}$ 2.09, написанного на немецком:

Hubert Partl <partl@mail.boku.ac.at>
Zentraler Informatikdienst der Universität für Bodenkultur Wien
Irene Hyna <Irene.Hyna@bmwf.ac.at>
Bundesministerium für Wissenschaft und Forschung Wien
Elisabeth Schlegl <noemail>
in Graz

Если вы интересуетесь немецким вариантом документа, вы можете найти его версию, которую Jörg Knappen обновил для $\text{\LaTeX}$ 2 ε , по адресу `CTAN:/tex-archive/info/lshort/german`.

В процессе подготовки этого документа я просил о рецензиях в `comp.text.tex`. Я получил множество откликов. Читатели помогли улучшить эту книжку исправлениями, предложениями и материалами. Они старались помочь мне довести документ до его нынешнего состояния. Я хотел бы искренне их всех поблагодарить. Естественно, все ошибки, которые вы найдете в этой книжке, — мои. Случайно попавшее сюда правильно написанное слово наверняка обязано своим появлением репликой от одного из нижеперечисленных.

Rosemary Bailey, Marc Bevand, Friedemann Brauer, Jan Busa, Markus Brühwiler, Pietro Braione, David Carlisle, José Carlos Santos, 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, 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, Alain Kessi, Christian Kern, Jörg Knappen, Kjetil Kjernsmo, Maik Lehradt, Rémi Letot, Johan Lundberg, Alexander Mai, Martin Maechler, Aleksandar S Milosevic, Henrik Mitsch, Claus Malten, Kevin Van Maren, Philipp Nagele, Lenimar Nunes de Andrade, Urs Oswald, Demerson Andre Polli, Maksym Polyakov Hubert Partl, John Reffling, Mike Ressler, Brian Ripley, Young U. Ryu, Bernd Rosenlecher, Chris Rowley, Risto Saarelna, Hanspeter Schmid, Craig Schlenter, Baron Schwartz, Christopher Sawtell, Geoffrey Swindale, Laszlo Szathmary, Boris Tobostras, Josef Tkadlec, Scott Veirs, Didier Verna, Fabian Wernli, Carl-Gustav Werner, David Woodhouse, Chris York, Fritz Zaucker, Rick Zacccone, и Mikhail Zotov.

Предисловие

L^AT_EX [1] — система верстки, ориентированная на производство научных математических документов высокого типографского качества. Система также вполне подходит для производства других видов документов, от простых писем до полностью сверстанных книг. L^AT_EX использует T_EX [2] в качестве своего механизма верстки.

Это краткое введение описывает L^AT_EX 2_ε и должно быть достаточно для большинства применений L^AT_EX. Для полного описания системы L^AT_EX читайте [1, 3].

Это введение разбито на пять глав:

Глава 1 рассказывает о базовой структуре документов L^AT_EX. Вы получите некоторое представление об истории L^AT_EX. После чтения этой главы вы должны получить грубое представление о работе L^AT_EX.

Глава 2 углубляется в детали верстки ваших документов. Она объясняет большинство важных команд и окружений L^AT_EX. После чтения этой главы вы сможете писать первые документы.

Глава 3 объясняет, как верстать формулы в L^AT_EX. Множество примеров поможет вам понять, как использовать эту, одну из самых сильных, сторону L^AT_EX. В конце этой главы вы найдете таблицы, перечисляющие большинство известных в L^AT_EX математических символов.

Глава 4 рассказывает про генерацию предметного указателя и библиографии и включение EPS графики. Она объясняет, как при помощи pdfL^AT_EX создавать документы в формате PDF, а также представляет несколько других полезных расширений, таких, как пакет XY-pic.

Глава 5 показывает, как использовать L^AT_EX для создания графики. Вместо рисования иллюстрации в каком-нибудь графическом редакторе, записи ее в файл и включения его в L^AT_EX, вы описываете иллюстрацию, а L^AT_EX рисует ее.

Глава 6 содержит потенциально опасные сведения о том, как менять стандартный макет документа. Она расскажет вам, как сделать красивый вывод L^AT_EX ужасным или потрясающим, в зависимости от ваших способностей.

Важно читать главы последовательно. В конце концов, книжка не такая уж большая. По мере чтения вам встретится большое количество примеров. Читайте их внимательно, поскольку многое в этой книжке объясняется именно в тексте примеров.

Л^AT_EX работает на большинстве компьютеров, начиная с IBM PC или Mac, и кончая большими системами UNIX или VMS. В многих университетских сетях система уже установлена и готова к работе. Информация о том, как использовать локальную установку Л^AT_EX, должна быть предоставлена в *Local Guide* [5]. Если у вас будут проблемы с началом работы, попросите о помощи того, кто предоставил вам эту книжку. Цель ее заключается *не* в том, чтобы обучить вас установке и настройке системы Л^AT_EX, а в том, чтобы научить, как писать ваши документы так, чтобы они могли быть обработаны Л^AT_EX.

Если вам понадобится любой, относящийся к Л^AT_EX, материал, поищите его на одном из ftp архивов Comprehensive T_EX Archive Network (СТАН). Его основной адрес — <http://www.ctan.org>. Все пакеты можно также получить с ftp-архива <ftp://www.ctan.org> и его зеркал по всему миру. Например, для США это — <ftp://ctan.tug.org>, для Германии — <ftp://ftp.dante.de>, для Великобритании — <ftp://ftp.tex.ac.uk>, для России — <ftp://ftp.radio-msu.net>. Если вы не в одной из этих стран, выберите ближайший к вам архив.

Далее в книге вы будете постоянно встречать ссылки на СТАН, обычно указывающие на полезные программные пакеты и документы. Вместо полной записи конкретных адресов будет писаться просто СТАН: и далее адрес в дереве файлов СТАН.

Если вы хотите иметь Л^AT_EX на вашем собственном компьютере, посмотрите на то, что доступно по адресу `systems`.

Если у вас есть мысли по поводу того, что сто́ит добавить, удалить или изменить в этом документе, пожалуйста, дайте мне знать. Я особенно заинтересован в откликах от новичков в Л^AT_EX на тему того, какие части введения легко понимаемы, и что можно объяснить лучше.

Tobias Oetiker <oetiker@ee.ethz.ch>

Department of Information Technology and
Electrical Engineering, Swiss Federal Institute of Technology

Текущая версия этого документа доступна по адресу [info/lshort](http://info.lshort1)¹.

¹Текущая версия данного перевода доступна по адресу <http://xtalk.msk.su/tex>, а также на СТАН. — Прим. перев.

Оглавление

Благодарности	iii
Предисловие	v
1 Это нужно знать	1
1.1 Названия	1
1.1.1 $\text{\TeX}$	1
1.1.2 $\text{\LaTeX}$	1
1.2 Основы	1
1.2.1 Автор, дизайнер и верстальщик	1
1.2.2 Дизайн макета	2
1.2.3 Преимущества и недостатки	2
1.3 Исходные файлы $\text{\LaTeX}$	4
1.3.1 Пробелы	4
1.3.2 Спецсимволы	4
1.3.3 Команды $\text{\LaTeX}$	5
1.3.4 Комментарии	5
1.4 Структура входного файла	6
1.5 Типичная сессия работы с $\text{\LaTeX}$	8
1.6 Макеты документов	9
1.6.1 Классы документов	9
1.6.2 Пакеты	9
1.6.3 Стили страницы	13
1.7 Встречающиеся типы файлов	13
1.8 Большие проекты	14
2 Верстка текста	17
2.1 Структура текста и языка	17
2.2 Разбиение на строки и страницы	19
2.2.1 Выровненные абзацы	19
2.2.2 Переносы	20
2.3 Специальные строки	21
2.4 Специальные буквы и символы	21
2.4.1 Знаки кавычек	21
2.4.2 Тире и дефисы	22

2.4.3	Тильда ($\sim$)	22
2.4.4	Знак градуса ($^{\circ}$)	22
2.4.5	Символ Евро (€)	22
2.4.6	Многоточие ($\dots$)	23
2.4.7	Лигатуры	23
2.4.8	Акценты и специальные символы	24
2.5	Поддержка иностранных языков	24
2.5.1	Поддержка португальского	27
2.5.2	Поддержка французского	27
2.5.3	Поддержка немецкого	28
2.5.4	Поддержка корейского	29
2.5.5	Поддержка кириллицы	31
2.6	Пробелы между словами	33
2.7	Заголовки, главы и разделы	33
2.8	Перекрестные ссылки	35
2.9	Сноски	36
2.10	Выделенные слова	36
2.11	Окружения	37
2.11.1	Список, перечисление и описание	37
2.11.2	Выравнивание влево, вправо и по центру	38
2.11.3	Цитаты и стихи	38
2.11.4	Буквальное воспроизведение	39
2.11.5	Таблицы	40
2.12	Плавающие объекты	42
2.13	Защита хрупких команд	45
3	Набор математических формул	47
3.1	Общие сведения	47
3.2	Группировка в математическом режиме	49
3.3	Составляющие математической формулы	49
3.4	Математические пробелы	53
3.5	Вертикально расположенный материал	54
3.6	Фантомы	56
3.7	Размер математического шрифта	56
3.8	Теоремы, законы,	57
3.9	Полужирные символы	59
3.10	Список математических символов	60
4	Специальные возможности	67
4.1	Включение Encapsulated POSTSCRIPT графики	67
4.2	Библиография	69
4.3	Указатели	70
4.4	Настраиваемые колонтитулы	71
4.5	Пакет verbatim	72

4.6	Загрузка и установка пакетов $\text{\LaTeX}$	73
4.7	Работа с $\text{pdf}\text{\LaTeX}$	74
4.7.1	PDF-документы для WWW	74
4.7.2	Шрифты	75
4.7.3	Использование графики	77
4.7.4	Гиперссылки	77
4.7.5	Проблемы со ссылками	80
4.7.6	Проблемы с закладками	80
4.8	Совместимость исходных текстов $\text{\LaTeX}$ и $\text{pdf}\text{\LaTeX}$	81
4.9	Создание презентаций при помощи pdfscreen	82
5	Генерация математической графики	85
5.1	Обзор	85
5.2	Окружение picture	86
5.2.1	Основные команды	86
5.2.2	Отрезки	87
5.2.3	Векторы	88
5.2.4	Окружности	89
5.2.5	Текст и формулы	90
5.2.6	Команды $\text{\backslash multiput}$ и $\text{\backslash linethickness}$	90
5.2.7	Овалы. Команды $\text{\backslash thinlines}$ и $\text{\backslash thicklines}$	91
5.2.8	Повторное использование блоков картинки	92
5.2.9	Квадратичные кривые Безье	93
5.2.10	Цепная линия	94
5.2.11	Скорость в специальной теории относительности	95
5.3	$\text{\Xypic}$	95
6	Настройка $\text{\LaTeX}$	99
6.1	Новые команды, окружения и пакеты	99
6.1.1	Новые команды	100
6.1.2	Новые окружения	101
6.1.3	Ваш собственный пакет	101
6.2	Шрифты и их размеры	102
6.2.1	Команды смены шрифта	102
6.2.2	Опасность!	104
6.2.3	Совет	105
6.3	Интервалы	105
6.3.1	Интервалы между строками	105
6.3.2	Форматирование абзацев	106
6.3.3	Горизонтальные интервалы	107
6.3.4	Вертикальные интервалы	108
6.4	Компоновка страницы	108
6.5	Еще о длинах	110
6.6	Блоки	111

6.7 Линейки и распорки	113
Литература	115
Предметный указатель	117

Список иллюстраций

1.1	Минимальный файл $\text{\LaTeX}$	7
1.2	Пример реалистичной журнальной статьи	7
4.1	Пример настройки <code>fancyhdr</code>	72
4.2	Пример входного файла <code>pdfscreen</code>	82
6.1	Пример пакета	102
6.2	Параметры компоновки страницы	109

Список таблиц

1.1	Классы документов	9
1.2	Опции классов документов	10
1.3	Некоторые из распространяемых с $\text{\LaTeX}$ пакетов	12
1.4	Предопределенные стили страницы $\text{\LaTeX}$	13
2.1	Акценты и специальные символы	24
2.2	Преамбула для португальских документов	27
2.3	Специальные команды для французского языка	28
2.4	Специальные символы немецкого языка	29
2.5	Болгарский, русский и украинский	32
2.6	Ключи размещения плавающего объекта	43
3.1	Акценты математического режима	60
3.2	Строчные греческие буквы	60
3.3	Прописные греческие буквы	60
3.4	Бинарные отношения	61
3.5	Бинарные операторы	61
3.6	Большие операторы	62
3.7	Стрелки	62
3.8	Ограничители	62
3.9	Большие ограничители	62
3.10	Прочие символы	63
3.11	Не-математические символы	63
3.12	Ограничители AMS	63
3.13	Буквы греческого и иврита AMS	63
3.14	Бинарные отношения AMS	64
3.15	Стрелки AMS	64
3.16	Отрицательные бинарные отношения и стрелки AMS	65
3.17	Бинарные операторы AMS	65
3.18	Прочие символы AMS	66
3.19	Математические алфавиты	66
4.1	Названия опций пакета <code>graphicx</code>	68
4.2	Примеры синтаксиса ключей указателя	70
6.1	Шрифты	103

6.2	Размеры шрифта	103
6.3	Абсолютные размеры шрифтов в стандартных классах	103
6.4	Математические шрифты	104
6.5	Единицы размерности в $\text{\TeX}$	107

1 Это нужно знать

Первая часть этой главы содержит краткий обзор философии и истории $\text{\LaTeX}$. Вторая часть главы фокусируется на основных структурах документов $\text{\LaTeX}$. После чтения этой главы вы должны иметь общее представление о том, как работает $\text{\LaTeX}$. В дальнейшем это поможет вам объединить всю новую информацию в единую картину.

1.1 Названия

1.1.1 $\text{\TeX}$

$\text{\TeX}$ — это компьютерная программа, созданная Дональдом Кнудом (Donald E. Knuth) [2]. Она предназначена для верстки текста и математических формул. Кнут начал писать $\text{\TeX}$ в 1977 году для изучения потенциала цифрового печатающего оборудования, которое начало появляться в это время, надеясь, в особенности, обратить тенденцию ухудшения типографского качества, которую он видел на примере его собственных книг и статей. $\text{\TeX}$, в том виде, в каком мы его сегодня используем, был выпущен в 1982 году с некоторыми добавлениями в 1989 (лучшая поддержка 8-битных символов и различных языков). $\text{\TeX}$ знаменит своей чрезвычайной стабильностью, работой на различных типах компьютеров и практически полным отсутствием ошибок. Номер версии $\text{\TeX}$ сходится к π и сейчас равен 3.14159.

$\text{\TeX}$ произносится как «тех». В среде ASCII $\text{\TeX}$ нужно писать как `TeX`.

1.1.2 $\text{\LaTeX}$

$\text{\LaTeX}$ — макропакет, позволяющий авторам верстать и печатать их работы с высоким типографским качеством, при помощи заранее определенных, профессиональных макетов. $\text{\LaTeX}$ был написан Leslie Lamport [1]. В качестве механизма для верстки он использует $\text{\TeX}$. Сейчас $\text{\LaTeX}$ поддерживает Frank Mittelbach.

$\text{\LaTeX}$ произносится как «лэйтех» или как «латех». Если вы ссылаетесь на $\text{\LaTeX}$ в ASCII окружении, пишите `LaTeX`. $\text{\LaTeX} 2_{\epsilon}$ пишется как `LaTeX2e`.

1.2 Основы

1.2.1 Автор, дизайнер и верстальщик

Для того, чтобы опубликоваться, авторы отдают свои рукописи в издательство. Затем один из дизайнеров издательства определяет макет документа (ширину столб-

цов, шрифты, интервалы выше и ниже заголовков и т.п.). Дизайнер записывает свои инструкции в рукописи и отдает ее верстальщику, который верстает книгу в соответствии с этими инструкциями.

Дизайнер-человек пытается понять, что автор имел в виду, когда писал свою рукопись. Он определяет заголовки глав, цитаты, примеры, формулы и прочее, исходя из своего профессионального опыта и из содержания рукописи.

В среде $\text{\LaTeX}$, $\text{\LaTeX}$ берет на себя роль дизайнера книги, используя $\text{\TeX}$ в качестве верстальщика. Но $\text{\LaTeX}$ — это всего лишь программа, и, следовательно, нуждается в более четких инструкциях. Автор должен предоставить дополнительную информацию, описывающую логическую структуру своей работы. Эта информация записывается в текст в виде «команд $\text{\LaTeX}$ ».

Это в корне отличается от WYSIWYG¹ подхода, принятого в большинстве современных текстовых процессоров, таких как *MS Word* или *Corel WordPerfect*. В этих приложениях авторы форматируют документ интерактивно в процессе набора текста на компьютере. В процессе работы они могут видеть на экране как будет выглядеть их работа, когда, в конце концов, она будет напечатана.

При использовании $\text{\LaTeX}$ обычно невозможно увидеть итоговую картину во время печатания текста. Ее, однако, можно посмотреть на экране после обработки файла $\text{\LaTeX}$. Затем можно внести исправления перед собственно печатью.

1.2.2 Дизайн макета

Типографский дизайн — это профессия. Неопытные авторы часто допускают серьезные ошибки форматирования, предполагая, что дизайн книги — это большей частью вопрос эстетики: «если документ выглядит художественно, значит, он хорошо отдизайнен». Но, так как документ предназначен для чтения, а не для вывешивания в картинной галерее, удобство его чтения и понимания гораздо более важны, нежели красота. Например:

- Размер шрифта и нумерация заголовков должны выбираться с тем, чтобы сделать структуру глав и разделов ясной для читателя.
- Строка должна быть достаточно короткой, чтобы не напрягать глаза читателя, и достаточно длинной для красивого заполнения страницы.

С WYSIWYG системами авторы часто производят эстетически приятные документы со слабо выраженной или невыдержанной структурой. $\text{\LaTeX}$ предотвращает такие ошибки форматирования, заставляя автора объявлять *логическую* структуру его документа. Затем уже $\text{\LaTeX}$ выбирает наиболее подходящий макет (раскладку) документа.

1.2.3 Преимущества и недостатки

Тема, часто обсуждаемая, когда люди из мира WYSIWYG встречаются с пользователями $\text{\LaTeX}$, — «преимущества $\text{\LaTeX}$ перед нормальными текстовыми процес-

¹What you see is what you get.

сорами», или наоборот. Лучшее, что вы можете сделать, когда начинается такая дискуссия, — это пригнуться, так как она часто выходит из-под контроля. Однако, иногда вы не можете уклониться . . .

Вот вам некоторое оружие. Основные преимущества $\text{\LaTeX}$ перед обычными текстовыми процессорами:

- Готовые профессионально выполненные макеты, делающие документы действительно выглядящими «как изданные».
- Удобно поддерживается верстка математических формул.
- Пользователю нужно выучить лишь несколько понятных команд, задающих логическую структуру документа. Ему практически никогда не нужно возиться собственно с макетом документа.
- Легко изготавливаются даже сложные структуры, типа примечаний, оглавлений, библиографий и прочее.
- Для решения многих типографских задач, не поддерживаемых напрямую базовым $\text{\LaTeX}$, есть свободно распространяемые дополнительные пакеты. Например, существуют пакеты для включения POSTSCRIPT-графики или для верстки библиографий в точном соответствии с конкретными стандартами. Многие из этих дополнительных компонент описаны в *The $\text{\LaTeX}$ Companion* [3].
- $\text{\LaTeX}$ поощряет авторов писать хорошо структурированные документы, так как именно так $\text{\LaTeX}$ и работает — путем спецификации структуры.
- $\text{\TeX}$, формирующее сердце $\text{\LaTeX}$ 2 ϵ , чрезвычайно мобилен и свободно доступен. Поэтому система работает практически на всех существующих платформах.

$\text{\LaTeX}$ имеет также и некоторые недостатки, но, кажется, мне трудно найти среди них заметные, хотя, я уверен, другие вам найдут их сотни ; -)

- Хотя предопределенные макеты имеют множество настраиваемых параметров, создание полностью нового макета документа не очень просто и занимает много времени.¹
- Очень сложно писать неструктурированные и неорганизованные документы.
- Ваша морская свинка может так до конца и не понять концепцию логической разметки, несмотря на видимые первые успехи.

¹Говорят, что это — одна из основных целей будущей системы $\text{\LaTeX}$ 3.

1.3 Исходные файлы $\text{\LaTeX}$

Исходными данными для $\text{\LaTeX}$ являются обычный текстовый файл в ASCII. Его можно создать в любом текстовом редакторе. Он содержит текст документа вместе с командами, указывающими $\text{\LaTeX}$, как верстать текст.

1.3.1 Пробелы

«Пустые» символы, такие, как пробел или табуляция, трактуются $\text{\LaTeX}$ одинаково, как «пробел». *Несколько последовательных* пустых символов трактуются как *один* «пробел». Пустые символы в начале строки обычно игнорируются, а единичный перевод строки воспринимается как «пробел».

Пустая строка между двух строк текста определяет конец абзаца. *Несколько* пустых строк трактуются так же, как *одна* пустая строка. Ниже приведен пример. Справа — текст из входного файла, слева — форматированный вывод.

Неважно, вставляете ли вы
один или несколько
пробелов между словами.

Пустая строка начинает
новый абзац.

Неважно, вставляете ли вы один или несколько
пробелов между словами.

Пустая строка начинает новый абзац.

1.3.2 Спецсимволы

Следующие символы являются зарезервированными символами, которые либо имеют в $\text{\LaTeX}$ специальное значение, либо имеются не во всех шрифтах. Если вы введете их в текст напрямую, то они обычно не напечатаются, а заставят $\text{\LaTeX}$ сделать что-нибудь, вами вовсе не предусмотренное.

\$ % ^ & _ { } ~ \

Как вы позже увидите, эти символы можно использовать в ваших документах, добавляя к ним префикс «\»:

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

\$ % ^ & _ { } ~

Прочие символы, как и многие, многие другие, можно набрать специальными командами в математических формулах или как акценты. Знак «\» *нельзя* вводить, добавляя перед ним еще один, так как эта команда (\\) используется для разрыва строки.¹

¹Вместо этого пользуйтесь командой `\backslash$`. Она дает ‘\’.

1.3.3 Команды $\text{\LaTeX}$

Команды $\text{\LaTeX}$ чувствительны к регистру и принимают одну из следующих двух форм:

- Они начинаются с символа backslash «\» и продолжаются именем, состоящим только из букв. Имена команд завершаются пробелом, цифрой или любой другой «не-буквой».
- Они состоят из «\» и ровно одного небуквенного символа.

$\text{\LaTeX}$ игнорирует пробелы после команд. Если вы хотите получить пробел после команды, вы должны поместить или «{ }» и пробел, или специальную команду пробела после имени команды. «{ }» не дает $\text{\LaTeX}$ игнорировать все пробелы после имени команды.

Я слышал, что Кнут разделяет
людей, работающих с $\text{\TeX}$
на $\text{\TeX}$ ников и $\text{\TeX}$ пертов.\\
Сегодня --- $\text{\today}$

Я слышал, что Кнут разделяет людей, работаю-
щих с $\text{\TeX}$ на $\text{\TeX}$ ников и $\text{\TeX}$ пертов.
Сегодня — 1 июня 2003 г.

Некоторые команды нуждаются в параметре, который должен быть задан между фигурными скобками «{ }» после имени команды. Некоторые команды поддерживают необязательные параметры, которые добавляются после имени команды в квадратных скобках «[]». Следующий пример использует некоторые команды $\text{\LaTeX}$. Не задумывайтесь над ними, они будут разъяснены позже.

Вы можете $\text{\texttt{\textbackslash textsl{положиться}}}$
на меня!

Вы можете *положиться* на меня!

Пожалуйста, начните новую
строчку прямо тут! $\text{\texttt{\textbackslash newline}}$
Спасибо!

Пожалуйста, начните новую строчку прямо тут!
Спасибо!

1.3.4 Комментарии

Когда в процессе обработки входного файла $\text{\LaTeX}$ встречается символ %, он игнорирует остаток текущей строки, возврат каретки и все пробелы в начале следующей строки.

Этим можно пользоваться для добавления в исходный файл замечаний, которые не будут выводиться на печать.

Это $\text{\texttt{Spercal\%}}$
 $\text{\texttt{ifragilist\%}}$
 $\text{\texttt{icexpialidocious}}$

Это $\text{\texttt{Spercalifragilisticexpialidocious}}$

Знаком `%` можно также пользоваться, чтобы разбить длинные строчки в тех местах, где не разрешаются пробелы или переводы строк.

Для более длинных комментариев можно также пользоваться окружением `comment`, предоставляемым пакетом `verbatim`. Это означает, что, для использования окружения `comment`, вы должны к преамбуле вашего документа добавить команду `\usepackage{verbatim}`:

```
Это~--- еще один
\begin{comment}
довольно глупый,
но полезный
\end{comment}
пример вставки комментариев
в ваш документ.
```

Это — еще один пример вставки комментариев в ваш документ.

Заметьте, что это не будет работать внутри сложных окружений, например, математики.

1.4 Структура входного файла

Когда $\text{\LaTeX} 2_\epsilon$ обрабатывает входной файл, он ожидает от него следования определенной структуре. Так, каждый входной файл должен начинаться с команды

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

Она указывает, документ какого типа вы собираетесь писать. После этого, вы можете включать команды, влияющие на стиль документа в целом, или загружать пакеты, добавляющие новые возможности в систему $\text{\LaTeX}$. Для загрузки такого пакета используется команда

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

Когда вся настройка закончена,¹ вы начинаете тело текста командой

```
\begin{document}
```

Теперь вы вводите текст с командами $\text{\LaTeX}$. В конце документа вы добавляете команду

```
\end{document}
```

Все, что следует после нее, $\text{\LaTeX}$ игнорирует.

Рис. 1.1 показывает содержимое минимального файла для $\text{\LaTeX} 2_\epsilon$. Несколько более сложный входной файл дан на рис. 1.2.²

¹Область между `\documentclass` и `\begin{document}` называется *преамбулой*.

²В данный пример вставлен также минимальный набор команд, необходимый для поддержки русского языка. Исторически сложились несколько пакетов русификации $\text{\LaTeX}$, здесь используется пакет `babel`, являющийся стандартным средством локализации $\text{\TeX}$. Если вы работаете на очень старой системе, возможно, вам нужно будет пользоваться другими установленными средствами. Проконсультируйтесь с вашим *Local Guide* [5] или с администратором. — Прим. перев.

```
1 \documentclass{article}
2 \usepackage[koi8-r]{inputenc}
3 \usepackage[russian]{babel}
4 \begin{document}
5 Краткость~--- сестра таланта.
6 \end{document}
```

Рис. 1.1: Минимальный файл $\text{\LaTeX}$

```
1 \documentclass[a4paper,11pt]{article}
2 \usepackage[russian]{babel}
3 \begin{document}
4 % определяем титул
5 \author{Б.~Тоботрас}
6 \title{Минимализм}
7 \frenchspacing
8 \begin{document}
9 % генерируем титул
10 \maketitle
11 % вставляем оглавление
12 \tableofcontents
13 \section{Начало}
14 Вот тут и начинается моя замечательная статья.
15 \section{Конец}
16 \ldots{} а тут она кончается.
17 \end{document}
```

Рис. 1.2: Пример реалистичной журнальной статьи

1.5 Типичная сессия работы с L^AT_EX

Можно ручаться, что вам уже не терпится попробовать маленький пример L^AT_EX-файла, приведенный на странице 7. Давайте попробуем. Сам по себе, L^AT_EX не включает графического интерфейса. Он — просто программа, обрабатывающая указанный входной файл. Некоторые дистрибутивы L^AT_EX включают графическую оболочку, где вы можете выбрать мышкой компиляцию указанного файла. В других системах вам нужно набирать команды в текстовом режиме. Предполагается, что на вашей машине L^AT_EX установлен и работает¹.

1. Отредактируйте или создайте входной файл L^AT_EX. Этот файл должен быть плоским ASCII текстом. В Unix любой редактор создаст вам именно это. В Windows вы должны убедиться, что вы сохраняете файл в формате ASCII, или *Plain text*. Выбирая имя для файла, добавьте расширение `.tex`.
2. Обработайте ваш входной файл L^AT_EX. Если все пройдет без ошибок, вы получите файл `.dvi`. Чтобы получить правильное оглавление и перекрестные ссылки, вам может понадобиться запустить L^AT_EX несколько раз. Если во входном файле есть ошибка, L^AT_EX укажет вам на нее и остановит обработку входного файла. Чтобы вернуться в командную строку, нажмите `ctrl-D`.

```
latex foo.tex
```

3. Теперь вы можете просмотреть файл `.dvi`. Это можно сделать по-разному. Можно посмотреть его на экране командой

```
xdvi foo.dvi &
```

Она работает только в среде Unix с X11. Если вы работаете под Windows, попробуйте `yap` (“yet another previewer”).

Вы также можете преобразовать файл DVI в POSTSCRIPT для печати или просмотра при помощи Ghostscript:

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

Если вам повезет, то в составе вашей системы будет программа `dvipdf`, которая может преобразовать ваши `.dvi`-файлы прямо в PDF.

```
dvipdf foo.dvi
```

¹Это обычно так в любой современной Unix-системе.

1.6 Макеты документов

1.6.1 Классы документов

Первое, что $\text{\LaTeX}$ должен знать при обработке входного файла, это тип создаваемого автором документа. Он задается командой `\documentclass`.

`\documentclass[опции]{класс}`

Здесь *класс* определяет тип создаваемого документа. Таблица 1.1 перечисляет классы документов, рассматриваемые в этом введении. В состав $\text{\LaTeX} 2_{\epsilon}$ входят дополнительные классы для других документов, включая письма и слайды. Параметр *опции* изменяет поведение класса документа. Опции должны разделяться запятыми. В таблице 1.2 перечислены самые употребительные опции стандартных классов документов.

Пример: Входной файл для документа $\text{\LaTeX}$ может начинаться строкой

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

она заставляет $\text{\LaTeX}$ набирать документ как *статью*, с базовым размером шрифта *одиннадцать пунктов* и форматировать документ для *двусторонней* печати на бумаге *формата A4*.

1.6.2 Пакеты

В процессе написания вашего документа, вы, вероятно, обнаружите, что в некоторых областях базовый $\text{\LaTeX}$ не сможет решить ваши проблемы. Если вы захотите включить в документ графику, цветной текст или исходный код программы из внешнего

Таблица 1.1: Классы документов

article	для статей в научных журналах, презентаций, коротких отчетов, программной документации, приглашений...
report	для более длинных отчетов, содержащих несколько глав, небольших книжек, диссертаций...
book	для настоящих книг
slides	для слайдов. Использует большие буквы без засечек. Вместо этого можно использовать <code>FoilTeX</code> . ^a

^a`macros/latex/contrib/supported/foiltex`

Таблица 1.2: Опции классов документов

<code>10pt, 11pt, 12pt</code>	Устанавливает размер основного шрифта документа. Если ни одна из этих опций не указана, подразумевается <code>10pt</code> .
<code>a4paper, letterpaper...</code>	Определяет размер листа. По умолчанию подразумевается <code>letterpaper</code> . Так же могут быть указаны <code>a5paper</code> , <code>b5paper</code> , <code>executivepaper</code> и <code>legalpaper</code> .
<code>fleqn</code>	Выключные формулы будут выровнены влево, а не отцентрированы.
<code>leqno</code>	Формулы нумеруются слева, а не справа.
<code>titlepage, notitlepage</code>	Указывает, должна начинаться новая страница после заголовка документа или нет. По умолчанию класс <code>article</code> не начинает новую страницу, а <code>report</code> и <code>book</code> — начинают.
<code>onecolumn, twocolumn</code>	Заставляет L ^A T _E X набирать документ в один столбец или в два столбца.
<code>twoside, oneside</code>	Выбирает одно- или двусторонний вывод. По умолчанию классы <code>article</code> и <code>report</code> используют односторонний вывод, класс <code>book</code> — двусторонний вывод. Заметьте, что опция <code>twoside</code> <i>не</i> заставляет ваш принтер на самом деле печатать с двух сторон.
<code>landscape</code>	Меняет положение страницы на ландшафтное.
<code>openright, openany</code>	Делает главы начинающимися или только на правой странице, или на первой доступной. Это не работает с классом <code>article</code> , так как он ничего не знает о главах. Класс <code>report</code> по умолчанию начинает главы на следующей странице, а класс <code>book</code> — на правой.

файла, вам нужно будет расширить возможности $\text{\LaTeX}$. Такие расширения называются пакетами. Пакеты активизируются командой

`\usepackage[опции]{пакет}`

Где *пакет* — это имя пакета, а *опции* — список ключевых слов, включающих специальные свойства пакета. Некоторые пакеты включены в основную поставку $\text{\LaTeX 2}_\epsilon$ (см. таблицу 1.3). Другие предоставляются отдельно. Дополнительная информация об установленных у вас пакетах может содержаться в *Local Guide* [5]. Основным источником информации о пакетах $\text{\LaTeX}$ — это *The $\text{\LaTeX}$ Companion* [3]. Она содержит описания сотен пакетов вместе с информацией о том, как писать ваши собственные расширения для $\text{\LaTeX 2}_\epsilon$.

Таблица 1.3: Некоторые из распространяемых с L^AT_EX пакетов

<code>doc</code>	Позволяет документировать программы на L ^A T _E X. Описан в <code>doc.dtx</code> ^a и в <i>The L^AT_EX Companion</i> [3].
<code>exscale</code>	Предоставляет масштабированные версии расширенных математических шрифтов. Описан в <code>ltxscale.dtx</code> .
<code>fontenc</code>	Указывает, какую кодировку шрифта должен использовать L ^A T _E X. Описан в <code>ltoutenc.dtx</code> .
<code>ifthen</code>	Предоставляет команды вида ‘если . . . , то выполнять . . . , иначе выполнять . . . ’. Описан в <code>ifthen.dtx</code> и в <i>The L^AT_EX Companion</i> [3].
<code>latexsym</code>	Чтобы подключить шрифт специальных символов L ^A T _E X, нужно использовать пакет <code>latexsym</code> . Описан в <code>latexsym.dtx</code> и в <i>The L^AT_EX Companion</i> [3].
<code>makeidx</code>	Предоставляет команды для генерации указателей. Описан в разделе 4.3 и в <i>The L^AT_EX Companion</i> [3].
<code>syntonly</code>	Обрабатывает документ, не печатая его.
<code>inputenc</code>	Позволяет указать входную кодировку, такую как ASCII, ISO Latin-1, ISO Latin-2, 437/850 IBM code pages, Apple Macintosh, Next, ANSI-Windows или определяемую пользователем. Описан в <code>inputenc.dtx</code> .

^aЭтот файл должен быть установлен на вашей системе, и вы можете получить `dvi` файл, напечатав `latex doc.dtx` в любом каталоге, где вы имеете права на запись. То же самое относится ко всем прочим файлам, упомянутым в этой таблице.

1.6.3 Стили страницы

$\text{\LaTeX}$ поддерживает три предопределенных комбинации верхнего колонтитула и нижнего колонтитула — так называемые стили страницы. Параметр *стиль* команды

```
\pagestyle{стиль}
```

определяет, какой из них использовать. Предопределенные стили страницы перечислены в таблице 1.4.

Таблица 1.4: Предопределенные стили страницы $\text{\LaTeX}$

plain	печатает номера страниц внизу страницы в середине нижнего колонтитула. Этот стиль установлен по умолчанию.
headings	печатает название текущей главы и номер страницы в верхнем колонтитуле каждой страницы, а нижний колонтитул остается пустым. (Этот стиль использован в данном документе.)
empty	делает и верхние, и нижние колонтитулы пустыми.

Возможно сменить стиль текущей страницы командой

```
\thispagestyle{стиль}
```

Описание того, как создавать ваши собственные стили колонтитулов, смотрите в *The $\text{\LaTeX}$ Companion* [3], а также в разделе 4.4 на странице 71.

1.7 Встречающиеся типы файлов

Работая с $\text{\LaTeX}$, вы вскоре начнете путаться в куче файлов с различными расширениями. Ниже перечислены различные типы файлов, используемые при работе с $\text{\TeX}$. Заметьте, что это не полный список расширений, но, если вы найдете не упомянутое расширение, которое считаете важным, — уведомите, пожалуйста, автора.

Следующие файлы генерируются, когда $\text{\LaTeX}$ обрабатывает входной файл:

- .dvi** Device Independent file (файл, не зависящий от устройства). Это — основной результат запуска $\text{\LaTeX}$. Содержимое его можно увидеть при помощи программы отображения DVI, или распечатать программой **dvips** или аналогичной.
- .log** Содержит детальный отчет о том, что происходило в последний прогон компиляции.

- `.toc` Хранит заголовки всех разделов. Читается в следующий проход компиляции и используется при генерации оглавления.
- `.lof` Аналог `.toc` для списка иллюстраций.
- `.lot` То же, для списка таблиц.
- `.aux` Еще один файл, передающий информацию между проходами компиляции. Кроме все прочего, используется для генерации перекрестных ссылок.
- `.idx` Если ваш документ содержит предметный указатель, `ЛATEX` помещает все слова для указателя в этот файл. Обработайте его программой `makeindex`. Подробнее смотрите в разделе 4.3 на странице 70.
- `.ind` Обработанный файл `.idx`, готовый ко включению в ваш документ при следующем проходе компиляции.
- `.ilg` Журнал работы `makeindex`.

1.8 Большие проекты

При работе с большими документами вам может быть удобно разделить входной файл на несколько частей. `ЛATEX` содержит две команды, которые помогают это делать.

`\include{файл}`

Эту команду можно использовать в теле документа, чтобы включить в него содержимое другого файла. Заметьте, что `ЛATEX` начнет новую страницу, прежде, чем обрабатывать материал, содержащийся в *файле*.

Вторая команда может использоваться только в преамбуле. Она указывает `ЛATEX` читать только некоторые из включенных (`\include`) файлов.

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

После выполнения в преамбуле документа этой команды будут выполняться только те команды `\include`, чьи имена файлов перечислены в аргументе команды `\includeonly`.

Команда `\include` начинает верстку включаемого текста на новой странице. Это удобно при использовании `\includeonly`, потому что границы страниц не будут двигаться, даже когда некоторые включаемые файлы пропускаются. Иногда это неудобно, и в этом случае вы можете использовать команду

`\input{filename}`

Она просто включает содержимое указанного файла.

Для быстрой проверки документа используйте пакет `syntonly`. Он заставляет $\text{\LaTeX}$ пробежать по документу, проверяя синтаксис и использование команд, но не генерируя выходного файла DVI. $\text{\LaTeX}$ в этом режиме работает быстрее, экономя ваше время. Синтаксис очень прост:

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

Когда вы захотите получить выходной файл, просто закомментируйте вторую строчку, добавив знак процента.

2 Верстка текста

После прочтения предыдущей главы вы должны представлять, из чего состоит документ $\text{\LaTeX}$. Вторая глава заполняет эту структуру всем необходимым для производства реальных материалов.

2.1 Структура текста и языка

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

Основная цель написания текста — передать читателю идеи, информацию или знания. Читатель лучше поймет текст, если эти идеи структурированы, и намного лучше увидит и почувствует эту структуру, если типографская форма отражает логическую и смысловую структуру содержания.

$\text{\LaTeX}$ отличается от других систем верстки в том, что вам нужно лишь задавать ему логическую и смысловую структуру текста. Он затем выбирает типографскую форму текста в соответствии с «правилами», заданными в файле класса документа и в различных стилевых файлах.

Самый важный элемент текста в $\text{\LaTeX}$ (и в наборе вообще) — абзац. Он называется «элементом текста», потому что абзац является той типографской формой, в которую облачается одна связная мысль или идея. В следующих разделах вы узнаете, как вы можете вызвать перевод строки, например, при помощи `\\` и начало абзаца, например, оставив пустую строку. Поэтому, если начинается новая мысль, должен начинаться новый абзац, а, если нет, — используйте переводы строки. Если вы сомневаетесь в вашем разбиении на абзацы, подумайте о вашем тексте как о носителе идей и мыслей. Если вы начали новый абзац, продолжая изложение той же мысли, — убирайте начало нового абзаца. Если на протяжении абзаца начинается совершенно новая мысль — разбивайте его на два.

Большинство людей совершенно не понимает важности разумного разбиения текста на абзацы. Многие не задумываются о значении разрыва абзацев или, особенно в $\text{\LaTeX}$, начинают абзацы, не зная этого. Последнюю ошибку особенно легко сделать, когда в текст включаются уравнения. Посмотрите на следующие примеры и разберитесь, почему иногда в них используются пустые строчки (разрывы параграфов), а иногда — нет. (Если вы еще не понимаете всех команд достаточно, чтобы понять примеры, прочтите, пожалуйста, эту и следующую главы и затем вернитесь к примерам снова.)

\ldots когда Эйнштейн
ввел свою формулу
\begin{equation}
e = m \cdot c^2 \ ; \ ,
\end{equation}
которая является самой
широко известной и самой
мало понятной физической
формулой.

... когда Эйнштейн ввел свою формулу

$$e = m \cdot c^2 , \quad (2.1)$$

которая является самой широко известной и са-
мой мало понятной физической формулой.

\ldots откуда следует
закон тока Кирхгофа:
\begin{equation}
\sum_{k=1}^n I_k = 0 \ ; \ .
\end{equation}

... откуда следует закон тока Кирхгофа:

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

Закон напряжения Кирхгофа
может быть выведен \ldots

Закон напряжения Кирхгофа может быть выве-
ден ...

\ldots который имеет
несколько преимуществ .

\begin{equation}
I_D = I_F - I_R
\end{equation}
является ядром совершенно
другой модели транзистора.
\ldots

... который имеет несколько преимуществ.

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

является ядром совершенно другой модели тран-
зистора. ...

Следующая, более маленькая, единица текста — предложение. В английском тексте после точки, завершающей предложение, ставится бóльший пробел, чем после точки, стоящей после сокращения. $\text{\LaTeX}$ старается определить, какую из них вы имели в виду. Если он ошибается, вы должны ему подсказать. Это объяснено далее в этой главе.

Структура текста охватывает даже отдельные части предложений. Большинство языков использует очень сложные правила пунктуации, но во многих языках (включая немецкий и английский) вы расставите почти все запятые на места, просто помня, что они означают: короткую остановку в потоке изложения. Если вы не уверены в своих запятых, прочтите предложение вслух, останавливаясь на каждой запятой. Если где-то это прозвучит странно, уберите эту запятую, если вы почувствуете необходимость остановки в другом месте, вставьте запятую.¹

Наконец, абзацы текста должны быть также логично структурированы на более высоком уровне, объединяясь в главы, разделы, и так далее. Однако, типографский

¹Все это мало применимо для русского, но я решил не выбрасывать этих абзацев из перевода, вдруг они кому-нибудь помогут писать по-английски. — *Прим. перев.*

эффект записи, например, `\section{Структура текста и языка}` обычно достаточно очевиден, чтобы было сразу ясно, как использовать эти высокоуровневые структуры.

2.2 Разбиение на строки и страницы

2.2.1 Выровненные абзацы

Книги часто верстаются так, чтобы все строчки имели одинаковую длину. $\text{\LaTeX}$ вставляет необходимые разрывы строк и пробелы между словами, оптимизируя форматирование абзаца как целого. При необходимости он также переносит слова, которые не помещаются на строке. От класса документа зависит то, как верстаются абзацы. Обычно каждый абзац начинается с красной строки, а дополнительного интервала между ними нет. Подробности смотрите в разделе 6.3.2.

В некоторых случаях может быть необходимо указать $\text{\LaTeX}$ разорвать строку:

`\\` или `\newline`

начинают новую строку, не начиная нового абзаца.

`\\*`

запрещает, кроме того, разрыв страницы после вставленного разрыва строки.

`\newpage`

начинает новую страницу.

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

разрывают строку, запрещают разрыв строки, разрывают страницу и запрещают разрыв страницы, соответственно. Необязательный аргумент n позволяет автору влиять на их действие. Он может быть равен числу от нуля до четырех. Устанавливая n в значение, меньшее 4, вы оставляете $\text{\LaTeX}$ возможность игнорировать вашу команду, если результат будет очень плох. Не путайте эти команды «разрыва» с командами «начала» (`new`). Даже если вы зададите команду «разрыва», $\text{\LaTeX}$ попытается выровнять правую границу страницы и общую высоту страницы, как описано в следующей секции. Если вы действительно хотите начать «новую строчку», используйте соответствующую команду.

$\text{\LaTeX}$ всегда пытается производить наилучшее из возможных разбиений строк. Если он не может найти способ разбить строки в соответствии со своими стандартами, он позволяет одной строке выступать из абзаца вправо. $\text{\LaTeX}$ затем выводит диагностику («`overfull hbox`») во время обработки входного файла. Чаще всего это

случается, когда $\text{\LaTeX}$ не может найти место для переноса слова.¹ Давая команду `\sloppy`, вы можете сказать, чтобы $\text{\LaTeX}$ несколько ослабил свои стандарты. Тогда он сможет предотвратить такие слишком длинные строки, увеличивая интервалы между словами — даже если конечный вывод будет не оптимален. В этом случае пользователь получит предупреждение («underfull hbox»). В большинстве случаев результат выглядит не очень хорошо. Команда `\fussy` действует в обратную сторону.

2.2.2 Переносы

$\text{\LaTeX}$ переносит слова, когда это необходимо. Если алгоритм переносов не находит правильных точек переноса, вы можете исправить положение, сказав $\text{\TeX}$ об исключении при помощи следующих команд.

Команда

```
\hyphenation{список слов}
```

вызывает перенос слов, перечисленных в ее аргументе, только в местах, отмеченных «-». Эта команда должна даваться в преамбуле входного файла, и должна содержать только слова, состоящие из обычных букв. Правила переноса запоминаются для языка, активного в момент обработки команды `\hyphenation`. Это значит, что, если вы поместите ее в преамбулу документа, она будет влиять на английские переносы. Если вы поместите ее после команды `\begin{document}`, используя при этом поддержку национальных языков, например, `babel`, то переносы будут запомнены для языка, активизированного при помощи `babel`.

Следующий пример разрешает переносы в слове «hyphenation», так же, как и в слове «Hyphenation», и запрещает переносы в словах «FORTRAN», «Fortran» и «fortran». В аргументе не допускаются специальные символы.

Пример:

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

Команда `\-` вставляет в слово выборочный перенос. Он также становится единственной разрешенной точкой переноса в этом слове. Эта команда в особенности полезна для слов, содержащих специальные символы (например, символы с акцентами), потому что $\text{\LaTeX}$ не переносит такие слова автоматически.

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

```
I think this is: supercalifragilisticexpialidocious
```

¹Хотя $\text{\LaTeX}$ и дает вам предупреждение, такие строки не всегда легко найти. Если вы в команде `\documentclass` используете опцию `draft`, такие строки будут отмечены толстой черной линией на правых полях.

Несколько слов можно удерживать вместе на одной строке командой

```
\mbox{текст}
```

Она будет в любом случае сохранять свой аргумент вместе.

Номер моего телефона скоро сменится.
Он будет \mbox{0116 291 2319}.

Параметр
\mbox{\emph{имя файла}} должен
содержать имя файла.

Номер моего телефона скоро сменится. Он будет
0116 291 2319.

Параметр *имя файла* должен содержать имя фай-
ла.

Команда \fbox аналогична команде \mbox, но вокруг ее содержимого будет нарисована видимая рамка.

2.3 Специальные строки

В некоторых предыдущих примерах вы уже встречали простые команды ЛАТЭХ для набора специальных текстовых строк:

Команда	Пример	Описание
\today	1 июня 2003 г.	Текущая дата на текущем языке
\TeX	TeX	Название вашей любимой системы верстки
\LaTeX	LaTeX	Название ЛАТЭХ
\LaTeXe	LaTeX 2 _ε	Текущая инкарнация ЛАТЭХ

2.4 Специальные буквы и символы

2.4.1 Знаки кавычек

Для набора кавычек вы *не должны* использовать знак ", как на пишущей машинке. При верстке существуют специальные знаки открывающей и закрывающей кавычек. В ЛАТЭХ используйте два знака ‘ в качестве открывающей кавычки и два знака ’ в качестве закрывающей.¹

‘Пожалуйста, нажмите клавишу ‘х’ ’

“Пожалуйста, нажмите клавишу ‘х’ ”

¹При верстке обычно требуются две пары кавычек, на случай «„кавычек“ в кавычках». При верстке на русском языке обычно вместо “таких” и ‘таких’ кавычек используют «эти» и „эти“. Какими командами они задаются — зависит от используемой русификации, но обычно это \flqq, \frqq (вместо этой пары часто используют лигатуры << и >>) и \glqq, \grqq. — Прим. перев.

2.4.2 Тире и дефисы

L^AT_EX знает про четыре вида тире. Три из них вы можете получить различным числом последовательных знаков -. Четвертое на самом деле не тире вовсе, а математический знак минус:

мальчик-с-пальчик, X-лучи\\
 страницы 13--67\\
 да~--- или нет?\\
 \$0\$, \$1\$ и \$-1\$

мальчик-с-пальчик, X-лучи
 страницы 13–67
 да — или нет?
 0, 1 и −1

Эти тире называются так: - дефис, -- короткое тире, --- длинное тире и \$-\$ знак минуса.

2.4.3 Тильда (~)

Тильда часто встречается в WWW-адресах. Для генерации ее в L^AT_EX можно использовать `\~`, но результат (`\~`) будет не тем, что вам нужен. Взамен попробуйте вот что:

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

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

2.4.4 Знак градуса (°)

Как напечатать знак градуса в L^AT_EX?

Температура уже
`$-30\,\sim{\circ}\mathrm{C}`\$.
 Скоро наступит сверхпроводимость.

Температура уже −30 °C. Скоро наступит сверхпроводимость.

2.4.5 Символ Евро (€)

В наше время, говоря о валюте, вам понадобится символ евро валюты. Многие современные шрифты содержат символ Евро. После загрузки пакета `textcomp` в преамбуле документа

```
\usepackage{textcomp}
```

вы можете пользоваться командой

```
\texteuro
```

для доступа к этому символу.

Если ваш шрифт не имеет собственного символа Евро, или вам не нравится, как он выглядит, у вас есть еще два выхода.

Первый — использование пакета `eurosym`. Он предоставляет официальный символ Евро:

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

Если вы предпочитаете официальному символ Евро, соответствующий вашему шрифту, вместо опции `official` используйте опцию `gen`.

Пакет `marvosym` включает множество разных символов, в том числе и Евро, доступный таким образом:

```
\EUR
```

Символы `\texteuro{}`, `\euro{}` и `\EUR{}` все выглядят по-разному.

Символы €, € и € все выглядят по-разному.

2.4.6 Многоточие (...)

На пишущей машинке точка или запятая занимают то же пространство, что и любая другая буква. Напечатанные в книге, эти символы занимают очень мало места и верстаются очень близко к предшествующей букве. Поэтому вы не можете ввести многоточие просто напечатав три точки, так как расстояние между ними будет неправильным. Кроме того, для многоточия есть специальная команда, называющаяся

```
\ldots
```

Не так ... а вот так: \\
Нью-Йорк, Токио, Будапешт, \ldots

Не так ... а вот так:
Нью-Йорк, Токио, Будапешт, ...

2.4.7 Лигатуры

Некоторые комбинации букв верстаются не просто набором разных букв друг за другом, а с использованием специальных символов.¹

ff fi fl ffi ... вместо ff fi fl ffi ...

¹Лигатуры широко используются при верстке текстов на английском языке. Насколько мне известно, при верстке на русском языке лигатур не используется. Однако, механизм лигатур TeX используется для генерации, например, кавычек и тире. — *Прим. перев.*

Эти так называемые лигатуры могут быть запрещены вставкой `\mbox{}` между двумя соответствующими буквами. Это может быть необходимо для слов, образованных из двух частей.

```
\Large He ‘‘shelfful’’\
a ‘‘shelf\mbox{ }ful’’
```

He “shelfful”
a “shelfful”

2.4.8 Акценты и специальные символы

Л^AT_EX поддерживает использование акцентов и специальных символов из многих языков. Таблица 2.1 показывает всевозможные акценты в применении к букве о. Понятно, что ее месте могут быть и другие буквы.

Чтобы поместить знак акцента над буквами i или j, точки над ними должны быть удалены. Это достигается набором `\i` и `\j`.

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

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

Таблица 2.1: Акценты и специальные символы

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

2.5 Поддержка иностранных языков

Если вам нужно писать документы на отличных от английского языках, то Л^AT_EX должен быть соответствующим образом сконфигурирован в двух пунктах:

1. Все генерируемые автоматически текстовые строки¹ должны быть переведены на другой язык. Для многих языков эти изменения достигаются использованием пакета `babel` (автор Johannes Braams).
2. $\text{\LaTeX}$ должен знать правила переноса для нового языка. Подключение правил переноса к $\text{\LaTeX}$ более сложно. Оно включает перестроение форматного файла с другими разрешенными образцами переноса. Ваш *Local Guide* [5] должен содержать больше информации об этом.
3. Специфичные для языка типографские правила. Скажем, для французского перед каждым символом двоеточия (:) должен стоять пробел.

Если ваша система уже соответствующим образом сконфигурирована, вы можете активизировать пакет `babel` добавлением команды

```
\usepackage[язык]{babel}
```

после команды `\documentclass`. Какие *языки* поддерживает ваша система, также может быть указано в *Local Guide* [5]. `Babel` автоматически активизирует правильные правила переноса для выбираемых вами языков. Если форматный файл вашего $\text{\LaTeX}$ не поддерживает переносов для выбранного языка, `babel` будет работать, но запретит переносы, что негативно скажется на внешнем виде документа.

Для некоторых языков `babel` вводит новые команды, упрощающие ввод специальных символов. Например, немецкий язык содержит множество диакритических знаков (äöü). С использованием `babel` вы можете вводить ö, печатая "o вместо \".o.

Если вы вызываете `babel`, указывая несколько языков,

```
\usepackage[язык1, язык2]{babel}
```

то для выбора текущего языка используйте команду

```
\selectlanguage{язык1}
```

Большинство современных компьютерных систем позволяет вам вводить специальные символы прямо с клавиатуры. $\text{\LaTeX}$ может обрабатывать такие символы при помощи пакета `inputenc`:

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

При использовании этого пакета вы должны понимать, что другие могут неправильно видеть ваши входные файлы на своем компьютере из-за использования другой кодировки. Например, немецкий акцент ä в системе IBM OS/2 кодируется как 132, а на некоторых Unix системах, использующих кодировку ISO-LATIN 1, — как

¹Содержание, Список иллюстраций, Библиография ...

228. Поэтому используйте эту возможность с осторожностью. В зависимости от вашей системы, могут использоваться, например, следующие кодировки:

Операционная система	Кодировка
Mac	<code>applemac</code>
Unix	<code>latin1</code>
Windows	<code>ansinew</code>
DOS, OS/2	<code>cp850</code>

Если вы пишете многоязычный документ с конфликтующими входными кодировками, можно переключиться на Unicode при помощи пакета `ucs`:

```
\usepackage{ucs}
\usepackage[utf8]{inputenc}
```

Это позволит вам генерировать файлы $\text{\LaTeX}$ в `utf8`, многобайтовой кодировке, где каждый символ кодируют от одного до четырех байт.

Иной случай — с кодировкой шрифта. Она определяет, в какой позиции в пределах шрифта $\text{\TeX}$ находится каждая буква. Множество входных кодировок можно отобразить на одну кодировку шрифта, что уменьшает число необходимых наборов шрифтов. Кодировки шрифтов обрабатываются пакетом `fontenc`:

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

где *кодировка* — требуемая кодировка шрифта. Можно одновременно загружать несколько кодировок.

Кодировка $\text{\LaTeX}$ по умолчанию — `OT1`, кодировка оригинального шрифта $\text{\TeX}$ Computer Modern. Она содержит только 128 символов 7-битного набора символов ASCII. Когда требуются акцентированные символы, $\text{\TeX}$ создает их, комбинируя нормальный символ с акцентом. Несмотря на то, что результирующая верстка выглядит прекрасно, этот подход не дает автоматическим переносам работать внутри слов, использующих акцентированные символы. Кроме того, некоторые знаки латинских языков не получается создать таким образом, не говоря уже о буквах не-латинских алфавитов, скажем, греческого или кириллицы.

Для борьбы с этими ограничениями было создано несколько 8-битных наборов шрифтов, похожих на CM. *Extended Cork* (EC) шрифты в кодировке `T1` содержат специальные символы для большинства акцентированных букв, используемых в европейских языках, основанных на латинице. Набор шрифтов `LN` содержит символы языков, основанных на кириллице. Ввиду большого числа кириллических знаков, они организованы в четыре кодировки шрифтов: `T2A`, `T2B`, `T2C` и `X2`.¹ Пакет `CB` содержит шрифты в кодировке `LGR` для набора греческих текстов.

¹Перечень языков, поддерживаемых каждой из этих кодировок можно найти в [11].

Таблица 2.2: Преамбула для португальских документов

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

Используя эти шрифты, вы можете улучшить или вообще сделать возможными переносы в не-английских документах. Еще одно преимущество использования новых СМ-подобных шрифтов — в том, что они предоставляют шрифты семейства СМ во всех яркостях, формах и оптически масштабированных размерах шрифта.

2.5.1 Поддержка португальского

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

Для включения переносов и смены автоматически генерируемого текста на португальский, пользуйтесь командой

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

Или, для работы с бразильскими текстами, ставьте в качестве языка **brazilian**.

Поскольку в португальском множество акцентированных букв, вам может быть удобно использовать

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

для того, чтобы корректно их вводить. Для работы переносов задайте

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

Преамбулу, необходимую для работы с португальским языком, смотрите в таблице 2.2. Заметьте, что мы здесь используем кодировку `latin1`, так что эта преамбула не будет работать на Mac или в DOS. Просто подставьте кодировку, правильную для вашей системы.

2.5.2 Поддержка французского

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

Несколько советов для создания в ЛАТ_ЭX французских документов. Загрузите

поддержку французского языка командой

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

Заметьте, что по историческим причинам название опции `babel` для французского языка — *frenchb* или *francais*, а не *french*.

Это включит французские переносы, если ваша $\text{\LaTeX}$ -система правильно настроена. Кроме того, весь автоматический текст будет тоже французским: `\chapter` печатает *Chapitre*, `\today` печатает текущую дату по-французски, и так далее. Становится доступным набор новых команд, позволяющих легче набирать французские тексты. Для пояснения смотрите таблицу 2.3.

Таблица 2.3: Специальные команды для французского языка

<code>\og guillemets \fg{}</code>	guillemets
<code>M\up{me}, D\up{r}</code>	$M^{\text{me}}, D^{\text{r}}$
<code>1\ier{ }, 1\iere{ }, 1\ieres{ }</code>	$1^{\text{er}}, 1^{\text{re}}, 1^{\text{res}}$
<code>2\ieme{ } 4\iemes{ }</code>	$2^{\text{e}} 4^{\text{es}}$
<code>\No 1, \no 2</code>	$N^{\circ} 1, n^{\circ} 2$
<code>20\~\degres C, 45\degres</code>	$20^{\circ} \text{C}, 45^{\circ}$
<code>\bsc{M. Durand}</code>	M. DURAND
<code>\nombre{1234,56789}</code>	1,234.567,89

Вы также заметите изменение вида списков. Чтобы полностью изучить изменения, вносимые `babel` при работе с французским, прогоните через $\text{\LaTeX}$ файл `frenchb.dtx` и прочтите полученный `frenchb.dvi`.

2.5.3 Поддержка немецкого

Несколько советов тем, кому нужно создавать в $\text{\LaTeX}$ немецкие документы. Загрузите поддержку немецкого языка командой

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

Это включит немецкие переносы, если ваша $\text{\LaTeX}$ -система правильно настроена. Кроме того, автоматически генерируемый текст станет немецким, например, “Chapter” станет “Kapitel”. Включается набор новых команд, ускоряющих набор немецкого текста, даже если не используется пакет `inputenc`. Для пояснения смотрите таблицу 2.4. С использованием `inputenc` все это становится не нужным, но ваш текст сможет использовать только одну кодировку.

Таблица 2.4: Специальные символы немецкого языка

"a	"a	"s	"s
"‘	"	"’	"
"< or \flqq	«	"> or \frqq	»
\flq	<	\frq	>
\dq	"		

В немецких книгах часто встречаются французские знаки кавычек («guillemets»). Немецкие типографы, однако, используют их по-другому. Цитата в немецкой книге обычно выглядит »так«. В немецкоговорящей части Швейцарии типографы используют «guillemets» так же, как и во Франции.

Большая проблема связана с командами типа `\flq`: в шрифтах OT1 (а они включаются по умолчанию) эти кавычки выглядят как математический символ “ $\ll$ ”, от чего у типографов будет истерика. Шрифты в кодировке T1, с другой стороны, имеют требуемые символы. Так что, если вам нужны такие кавычки, убедитесь, что включили кодировку T1 (`\usepackage[T1]{fontenc}`).

2.5.4 Поддержка корейского¹

Используя $\text{\LaTeX}$ для набора корейского языка, нам нужно разрешить три проблемы:

1. Нужно научиться редактировать входные файлы на корейском. Они должны быть в формате плоского текста, но, поскольку символы корейского языка не входят в набор US-ASCII, они будут странно выглядеть в стандартных редакторах ASCII. Две самых популярных кодировки корейских текстовых файлов — EUK-KR и ее совместимое расширение, используемое в корейской системе MS Windows, CP949/Windows-949/УНС. Эти кодировки включают ASCII как подмножество, аналогично кодировкам ISO-8859-*x*, EUC-JP, Shift_JIS и Big5. С другой стороны, слоги Hangul, знаки Hanja (используемые в Корее китайские символы), Hangul Jamos, хиракана, катакана, греческие, кириллица и прочие символы из KS X 1001 представляются двумя байтами. Первый байт имеет установленным старший бит. До середины 90-х годов установка корейского окружения в нелокализованной ОС требовала значительных усилий и времени. Вы можете просмотреть сильно устаревший <http://jshin.net/faq>, чтобы получить представление об этих усилиях. В наше время все три основные операционные

¹Рассматриваются несколько моментов, касающихся корейских пользователей $\text{\LaTeX}$. Этот раздел написал Karnes Kim от имени корейской команды переводчиков lshort. Раздел переведен на английский Shun Jungshik и сокращен Tobi Oetiker.

системы (Mac OS, Unix, Windows) включают вполне приличную многоязыковую поддержку и возможности интернационализации, так что редактирование корейских текстов уже не является проблемой, даже в нелокализованной системе.

2. $\text{\TeX}$ и $\text{\LaTeX}$ были изначально созданы для письменностей, имеющих в алфавите не более 256 символов. Для того, чтобы они работали с языками, где гораздо больше символов, например, корейским¹ и китайским, был разработан механизм подшифтов. Он делит единый шрифт CJK с тысячами или десятками тысяч знаков на набор подшифтов с 256 знаками в каждом. Для корейского языка есть три широко используемых пакета: $\text{\HLaTeX}$ (автор Un Koanghi), $\text{\hLaTeXr}$ (автор Cha Jaechoon) и пакет CJK (автор Werner Lemberg).² $\text{\HLaTeX}$ и $\text{\hLaTeXr}$ специфичны для корейского языка и предлагают корейскую локализацию, помимо шрифтовой поддержки. Оба они обрабатывают входные файлы в кодировке EUK-KR. $\text{\HLaTeX}$ может обрабатывать файлы в кодировках CP949/Windows-949/УНС и UTF-8, если используется вместе с Λ , Ω .

Пакет CJK не специфичен для корейского языка. Он обрабатывает входные файлы в UTF-8, а также в различных кодировках CJK, включая EUK-KR и CP949/Windows-949/УНС. Его можно использовать для набора документов с многоязычным содержанием (особенно китайским, японским и корейским). Пакет CJK не содержит корейской локализации, аналогичной той, что есть в $\text{\HLaTeX}$, и также не содержит такого множества специальных корейских шрифтов.

3. Конечная цель использования программ набора, включая $\text{\TeX}$, — получить документы, сверстанные «эстетически» удовлетворительно. Можно утверждать,

¹Корейский алфавит Hangul — буквенная письменность с 14 основными согласными и 10 основными гласными (Jamos). В отличие от латиницы или кириллицы, отдельные символы должны быть размещены в прямоугольных блоках примерно такого же размера, как и китайские. Каждый блок представляет слог. Из этого небольшого количества гласных и согласных можно сформировать бесконечное число слогов. Однако, современные орфографические стандарты (как в Южной Корее, так и в Северной) налагают некоторые ограничения на формирование этих блоков. Следовательно, существует конечное число орфографически правильных слогов. Корейские кодировки символов определяют индивидуальные коды для каждого из этих слогов (KS X 1001:1998 и KS X 1002:1992). Таким образом, Hangul, будучи буквенным, трактуется, тем не менее, так же, как письменности Китая и Японии, с тысячами идеографических знаков. Стандарт ISO 10646/Unicode предлагает оба способа представления Hangul, используемых в современной Корее, кодируя, в дополнение ко всем орфографически разрешенным слогам Hangul (<http://www.unicode.org/charts/PDF/UAC00.pdf>), еще и Conjoining Hangul Jamos (алфавит: <http://www.unicode.org/charts/PDF/U1100.pdf>). Одна из самых устрашающих задач в наборе корейского текста в $\text{\LaTeX}$ и аналогичных системах — поддержка средневековых корейских (а, возможно, и будущих корейских) слогов, которые можно представить только комбинацией знаков Jamos в Unicode. Есть надежда, что будущие варианты $\text{\TeX}$, например, Ω и Λ в конце концов предоставят решение для этой задачи, так что корейские лингвисты и историки смогут сбежать из MS Word, где уже есть довольно неплохая поддержка средневекового корейского.

²Их можно получить по адресам language/korean/HLaTeX/, language/korean/CJK/ и <http://knot.kaist.ac.kr/htex/>

что наиболее важный элемент верстки — это набор хорошо спроектированных шрифтов. Дистрибутив $\text{H}\text{A}\text{T}\text{E}\text{X}$ включает $\text{U}\text{N}\text{C}\ \text{P}\text{O}\text{S}\text{T}\text{S}\text{C}\text{R}\text{I}\text{P}\text{T}$ шрифты десяти различных семейств и $\text{T}\text{r}\text{u}\text{e}\text{T}\text{y}\text{p}\text{e}$ шрифты Munhwabu¹ пяти различных семейств. Пакет CJK работает с набором шрифтов, использовавшихся в старых версиях $\text{H}\text{A}\text{T}\text{E}\text{X}$, а также может использовать $\text{T}\text{r}\text{u}\text{e}\text{T}\text{y}\text{p}\text{e}$ шрифт cyberbit компании Bitstream.

Для загрузки пакета $\text{H}\text{A}\text{T}\text{E}\text{X}$ поместите в преамбуле следующее:

```
\usepackage{hangul}
```

Эта команда включает корейскую локализацию. Заголовки глав, разделов, подразделов, оглавления и перечня иллюстраций будут переведены на корейский, и форматирование документа изменится так, чтобы следовать принятым в Корее правилам. Пакет также предоставляет автоматический «выбор частиц». В корейском языке существуют пары постфиксных частиц, грамматически эквивалентных, но отличных по форме. Какая частица из пары корректна, зависит от того, заканчивается ли предыдущий слог на гласную или согласную. (На самом деле, правило чуть сложнее, но общая идея такова.) Корейцам не составляет никакого труда выбрать правильную частицу, однако невозможно заранее определить, какую частицу использовать в ссылках и других местах с автоматически генерируемым текстом, меняющимся при редактировании документа. Вставлять при каждом изменении текста правильную частицу везде вручную — решение совершенно ужасное. $\text{H}\text{A}\text{T}\text{E}\text{X}$ автоматизирует этот процесс.

Если вам не нужна корейская локализация, а вы просто хотите набрать корейский текст, поместите взамен в преамбулу следующее:

```
\usepackage{hfont}
```

Полная и подробная информация о верстке корейского текста при помощи $\text{H}\text{A}\text{T}\text{E}\text{X}$ приведена в *Руководстве по $\text{H}\text{A}\text{T}\text{E}\text{X}$* . Смотрите веб-сайт группы корейских пользователей TEX (KTUG) по адресу <http://www.ktug.or.kr/>. Существует также корейский перевод настоящего документа.

2.5.5 Поддержка кириллицы

Автор: Maksym Polyakov <polyama@myrealbox.com>

Пакет `babel`, начиная с версии 3.7, включает поддержку кодировок $\text{T}\text{2}^*$ и поддержку верстки болгарских, русских и украинских текстов с использованием кириллических символов.

Поддержка кириллицы основана на стандартных механизмах $\text{L}\text{A}\text{T}\text{E}\text{X}$: пакетах `fontenc` и `inputenc`. Но, если вам нужно использовать кириллицу в математическом

¹Министерство культуры Кореи

режиме, загрузите до пакета `fontenc` пакет `mathtext`:¹

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

Вообще говоря, `babel` автоматически выберет кодировку шрифтов по умолчанию. Для трех перечисленных языков это будет **T2A**. Однако, документы не ограничены только одной кодировкой шрифта. Для многоязыковых документов, использующих языки, базирующиеся на кириллице и латинице, имеет смысл явно подключить латинскую кодировку шрифтов. `babel` позаботится о переключении на правильную кодировку, когда вы меняете язык в документе.

В дополнение ко включению переносов, трансляции автоматически генерируемых строк и активации некоторых специфичных для языка типографских правил (например, `\frenchspacing`), `babel` предоставляет несколько команд для верстки в соответствии со стандартами болгарского, русского или украинского языков.

Для всех трех языков предоставляется специфическая пунктуация: кириллическое тире для теста (оно слегка уже латинского тире и окружено крошечными пробелами), тире для прямой речи, кавычки и команды управления переносами. Смотрите таблицу 2.5.

Таблица 2.5: Дополнительные определения, вводимые опциями `babel` для болгарского, русского и украинского языков

"	запрещает лигатуру в этой позиции.
"-	дефис, разрешающий переносить остаток слова.
"---	кириллическое тире в обычном тексте.
"--~	кириллическое тире в составных фамилиях.
"--*	кириллическое тире для обозначения прямой речи.
"	как , но не генерирует знака дефиса (для составных слов с дефисом, например, x-y).
"~	отмечает составное слово без разрыва.
"=	отмечает составное слово с разрывом, разрешая перенос в словах-компонентах.
" ,	пробел для инициалов с разрешением переноса в следующей за ними фамилии.
"‘	немецкая левая двойная кавычка (выглядит так: „).
"’	немецкая правая двойная кавычка (выглядит так: “).
"<	французская левая двойная кавычка (выглядит так: <<).
">	французская правая двойная кавычка (выглядит так: >>).

¹Если вы пользуетесь пакетами $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$, загружайте их также до пакетов `fontenc` и `babel`.

Русская и украинская опции пакета `babel` вводят команды `\Asbuk` и `\asbuk`, которые работают аналогично командам `\Alph` и `\alph`, но генерируют прописные и строчные буквы русского или украинского алфавитов (в зависимости от текущего языка документа). Болгарская опция `babel` вводит команды `\enumBul` и `\enumLat` (`\enumEng`), которые заставляют `\Alph` и `\alph` генерировать буквы либо болгарского, либо латинского (английского) алфавитов. По умолчанию генерируются болгарские буквы.

2.6 Пробелы между словами

Для получения ровного правого края вывода $\text{\LaTeX}$ вставляет различные интервалы между словами. В конце предложения он вставляет слегка больший интервал, делая текст более читабельным.¹ $\text{\LaTeX}$ предполагает, что предложения заканчиваются точками, вопросительными или восклицательными знаками. Если точка следует за буквой в верхнем регистре, она не считается концом предложения, так как точки после букв верхнего регистра обычно используются для сокращений.

Любое исключение из этих предположений должно быть явно оговорено автором. Знак «\» перед пробелом дает в результате пробел, который не будет увеличен. Знак «~» дает пробел, который не может увеличиться и который, кроме того, запрещает разрыв строки. Команда `\@` перед точкой указывает, что эта точка заканчивает предложение, несмотря на то, что стоит за буквой верхнего регистра.

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

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

Дополнительный пробел после точек можно запретить командой

`\frenchspacing`

которая указывает $\text{\LaTeX}$ не вставлять пробела после точки более, чем после обычных символов. Это обычно для языков, отличных от английского, за исключением библиографий. Если вы используете `\frenchspacing`, команда `\@` не нужна.

2.7 Заголовки, главы и разделы

Чтобы помочь читателю ориентироваться в вашей работе, вы должны разделять ее на главы, разделы и подразделы. $\text{\LaTeX}$ поддерживает это специальными командами, принимающими в качестве аргумента заголовки раздела. Ваше дело — использовать их в надлежащем порядке.

Класс `article` включает следующие команды секционирования:

¹В соответствии с традициями верстки, принятыми в английском языке. — *Прим. перев.*

```

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

```

В классах `report` и `book` вы можете использовать дополнительную команду `\chapter{...}`.

Если вы хотите разбить ваш документ на части без изменения нумерации разделов/глав, используйте команду `\part{...}`.

Так как глав (chapters) в классе `article` нет, то статьи довольно легко добавлять в книгу в качестве глав. Интервалы между разделами, нумерация и размер шрифта заголовков устанавливаются $\text{\LaTeX}$ автоматически.

Две из команд секционирования — особенные:

- Команда `\part` не влияет на последовательность нумерования глав.
- Команда `\appendix` аргумента не имеет. Она просто начинает нумеровать главы буквами вместо цифр.¹

$\text{\LaTeX}$ создает оглавление, беря заголовки разделов и номера страниц из предыдущего цикла компиляции документа. Команда

```
\tableofcontents
```

вставляет оглавление в то место, где она вызвана. Чтобы получить правильное оглавление, новый документ должен быть обработан $\text{\LaTeX}$ дважды. В особых случаях может быть необходим и третий проход. Когда это потребуется, $\text{\LaTeX}$ вас предупредит.

Все вышеперечисленные команды секционирования существуют также в вариантах со звездочкой. Такой вариант получается добавлением `*` к имени команды. Они генерируют заголовки разделов, которые не нумеруются и не включаются в оглавление. Например, команда `\section{Справка}` становится `\section*{Справка}`.

Обычно заголовки разделов появляются в оглавлении точно в том же виде, в каком они вводятся в тексте. Иногда это невозможно из-за того, что заголовок слишком длинен для оглавления. Элемент оглавления может в этом случае указываться необязательным аргументом перед собственно заголовком.

```

\chapter[Заголовок для оглавления]{Это~--- длинный,
длинный и очень нудный заголовок, появляющийся в тексте}

```

¹В классе `article` меняется нумерация разделов.

Титульный лист документа в целом генерируется при помощи команды

`\maketitle`

Его содержимое должно быть определено командами

`\title{...}, \author{...} и \date{...}`

до момента вызова `\maketitle`. Аргумент команды `\author` может содержать несколько имен, разделенных командами `\and`.

Пример некоторых из упомянутых команд может быть найден на иллюстрации 1.2 на странице 7.

Помимо описанных выше команд секционирования, $\text{\LaTeX}$ 2_ε вводит три дополнительных команды для использования с классом `book`. Они полезны для деления вашей публикации. Команды изменяют заголовки глав и нумерацию страниц так, как это ожидается от книги:

`\frontmatter` должна быть самой первой командой после `\begin{document}`. Она переключает нумерацию страниц на использование римских цифр. Для вводной части часто используют команды секционирования со звездочками (например, `\chapter*{Предисловие}`), чтобы $\text{\LaTeX}$ не нумеровал эти разделы.

`\mainmatter` указывается сразу перед первой главой книги. Она включает нумерацию страниц цифрами и сбрасывает счетчик страниц.

`\appendix` отмечает начало дополнительного материала в вашей книге. После этой команды главы будут нумероваться буквами.

`\backmatter` указывается перед самыми последними частями книги, например, перед библиографией и предметным указателем. В стандартных классах документов видимого эффекта она не имеет.

2.8 Перекрестные ссылки

В книгах, отчетах и статьях часто встречаются перекрестные ссылки на иллюстрации, таблицы и отдельные части текста. Для этого $\text{\LaTeX}$ предоставляет следующие команды:

`\label{метка}, \ref{метка} и \pageref{метка}`

где *метка* — выбранный пользователем идентификатор. $\text{\LaTeX}$ заменяет `\ref` номером раздела, подраздела, иллюстрации, таблицы или уравнения, где была использована соответствующая команда `\label`. `\pageref` печатает номер страницы, соответ-

ствующей команде `\label`.¹ Так же, как и в случае с заголовками разделов, здесь также используются номера из предыдущего прохода.

Ссылка на этот
раздел `\label{sec:this}`
выглядит так: <<см.
раздел `\ref{sec:this}` на
стр. `\pageref{sec:this}`.>>

Ссылка на этот раздел выглядит так: «см. раздел 22 на стр. 36.»

2.9 Сноски

Команда

```
\footnote{текст сноски}
```

печатает сноску внизу текущей страницы. Сноски всегда должны помещаться после слова или предложения, к которым они относятся. В русском языке сноски, относящиеся к предложению, должны ставиться сразу перед точкой или запятой².

Пользователи `\LaTeX`
часто употребляют
сноски. `\footnote{%
Это --- сноска.}`

Пользователи `\LaTeX` часто употребляют сноски.^a

^aЭто — сноска.

2.10 Выделенные слова

В рукописи, напечатанной на машинке, **важные слова выделяются подчеркиванием**:

```
\underline{текст}
```

В печатных изданиях, однако, эти слова выделяются *курсивом*. Команда для переключения на шрифт *выделения* называется

```
\emph{текст}
```

Ее аргументом является текст для выделения. Что на самом деле делает эта команда, зависит от контекста:

¹Заметьте, что эти команды не знают, на что именно они ссылаются. `\label` просто сохраняет последний автоматически генерируемый номер.

²Заметьте, что сноски отвлекают читателя от основного текста документа. Все всегда читают сноски, потому что мы — любопытные создания. Поэтому старайтесь включать все, что вы хотите сказать, в основной текст документа.

Если вы используете
`\emph{выделение в уже`
 выделенном тексте, то
`\LaTeX{}` использует
`\emph{прямой}` шрифт.}

Если вы используете *выделение в уже выделенном тексте*, то *LaTeX* использует прямой шрифт.

Отметьте отличие между командами *выделения* и смены *шрифта*:

`\textit{Вы можете также`
`\emph{выделить} текст,`
 набрав его курсивом,
`\textsf{шрифтом без`
`\emph{засечек}}`
`\texttt{или в стиле`
`\emph{пишущей машинки}}.`

Вы можете также выделить текст, набрав его курсивом, шрифтом без засечек или в стиле пишущей машинки.

2.11 Окружения

Для верстки специальных видов текста *LaTeX* определяет множество окружений для разных типов форматирования:

```
\begin{название} текст \end{название}
```

где *название* определяет окружение. Окружения можно вызывать внутри окружений, соблюдая порядок вызова и возврата:

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

В следующих разделах рассказывается обо всех важных окружениях.

2.11.1 Список, перечисление и описание

Окружение `itemize` подходит для простых списков, окружение `enumerate` — для нумерованных списков, а окружение `description` — для описаний.

```

\flushleft
\begin{enumerate}
\item Вы можете как угодно
смешивать окружения списков:
\begin{itemize}
\item Но это может смотреться
глупо.
\item[-] С минусом.
\end{itemize}
\item Поэтому помните:
\begin{description}
\item[Глупые] вещи не станут
умнее от помещения в список.
\item[Умные] вещи, однако,
вполне можно представить
списком.
\end{description}
\end{enumerate}

```

1. Вы можете как угодно смешивать окружения списков:
 - Но это может смотреться глупо.
 - С минусом.
2. Поэтому помните:

Глупые вещи не станут умнее от помещения в список.

Умные вещи, однако, вполне можно представить списком.

2.11.2 Выравнивание влево, вправо и по центру

Окружения `flushleft` и `flushright` форматируют абзацы, выровненные влево или вправо. Окружение `center` дает центрированный текст. Если вы не используете `\\` для указания разрывов строк, `LaTeX` определит их автоматически.

```

\begin{flushleft}
Этот текст\\ выровнен влево.
\LaTeX{} не старается сделать
все строки одинаковой длины.
\end{flushleft}

```

Этот текст
выровнен влево. `LaTeX` не старается сделать все
строки одинаковой длины.

```

\begin{flushright}
Этот текст\\ выровнен вправо.
\LaTeX{} не старается сделать
все строки одинаковой длины.
\end{flushright}

```

Этот текст
выровнен вправо. `LaTeX` не старается сделать все
строки одинаковой длины.

```

\begin{center}
В центре\\Земли
\end{center}

```

В центре
Земли

2.11.3 Цитаты и стихи

Окружение `quote` полезно для цитат, важных фраз и примеров.

Типографское правило для
длины строки:

```
\begin{quote}
  Обычно строки должны
  содержать не больше
  66~символов.
\end{quote}
```

Поэтому `\LaTeX{}` делает такими
широкими поля страниц. Поэтому
в газетах часто применяют набор
в несколько колонок.

Типографское правило для длины строки:

Обычно строки должны содержать не
больше 66 символов.

Поэтому `\LaTeX` делает такими широкими поля
страниц. Поэтому в газетах часто применяют на-
бор в несколько колонок.

Существуют еще два похожих окружения: `quotation` и `verse`. Окружение `quotation` полезно для более длинных цитат, охватывающих несколько абзацев, потому что оно начинает абзацы с красной строки. Окружение `verse` используют для стихов, где важны разрывы строк. Строки разделяются при помощи `\\` в конце строки и пустой строки после каждой строфы.

Я знаю только одно английское
стихотворение наизусть: про
Шалтая-Болтая:
`\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}`

Я знаю только одно английское стихотворение на-
изусть: про Шалтая-Болтая:

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 Буквальное воспроизведение

Текст, заключенные между `\begin{verbatim}` и `\end{verbatim}` будет напрямую напечатан, как набранный на пишущей машинке, со всеми пробелами и возвратами каретки, без выполнения каких бы то ни было команд `\LaTeX`.

Внутри абзаца аналогичную функцию выполняет команда

```
\verb+текст+
```

Здесь «+» — это только пример символа-ограничителя. Вы можете использовать любой символ, кроме букв, «*» или пробела. Многие примеры на `\LaTeX` в этом буклете набраны этой командой.

Команда `\verb|\ldots| \ldots`

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

Команда `\ldots` ...

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

```
\begin{verbatim*}
вариант  окружения
verbatim со
звездочкой выделяет
пробелы в тексте
\end{verbatim*}
```

```
вариант окружения
verbatim со
звездочкой выделяет
пробелы в тексте
```

Команду `\verb` тоже можно использовать аналогичным образом со звездочкой:

```
\verb*|вот так :-)|
```

```
вот так :-)
```

Окружение `verbatim` и команду `\verb` нельзя использовать внутри параметров других команд.

2.11.5 Таблицы

Окружение `tabular` используют для верстки таблиц, возможно, с горизонтальными и вертикальными линиями. $\text{\LaTeX}$ автоматически определяет ширину столбцов.

Аргумент *спецификация* команды

```
\begin{tabular}[позиция]{спецификация}
```

определяет формат таблицы. Используйте `l` для столбца текста, выровненного влево, `r` для текста, выровненного вправо и `c` для центрированного текста, `p{ширина}` для столбца, содержащего выровненный текст с переносом строк, и `|` для вертикальной линии.

Позиция определяет вертикальное положение всего табличного окружения: `t`, `b` и `c` означают выравнивание по верхнему краю, нижнему краю или по центру окружения.

Внутри окружения `tabular` знак «&» переходит к следующему столбцу, команда `\\` начинает новую строку, а `\hline` вставляет горизонтальную линию. Вы можете добавлять неполные линии при помощи команды `\cline{j-i}`, где `j` и `i` — номера столбцов, над которыми должна проходить линия.

```
\begin{tabular}{|r|l|}
\hline
7C0 & шестнадцатеричное \\
3700 & & восьмеричное \\
11111000000 & & двоичное \\
\hline \hline
1984 & десятичное \\
\hline
\end{tabular}
```

7C0	шестнадцатеричное
3700	восьмеричное
11111000000	двоичное
1984	десятичное

```
\begin{tabular}{|p{4.7cm}|}
\hline
Добро пожаловать в абзац в
рамочке. Надеемся, вам всем
тут понравится.\\
\hline
\end{tabular}
```

Добро пожаловать в абзац в рамочке. Надеемся, вам всем тут понравится.
--

Разделитель столбцов можно задать конструкцией `@{...}`. Эта команда удаляет пробел между столбцами и заменяет его на то, что включено в фигурные скобки. Одно из частых использований этой команды показано ниже, при рассказе о проблеме выравнивания по десятичной точке. Другое возможно использование — для подавления ведущего пробела в таблице при помощи `@{}`:

```
\begin{tabular}{@{} l @{}}
\hline
нет ведущего пробела\\
\hline
\end{tabular}
```

нет ведущего пробела

```
\begin{tabular}{l}
\hline
ведущий пробел слева и справа\\
\hline
\end{tabular}
```

ведущий пробел слева и справа

Поскольку встроенный способ выравнивать числовые столбцы по десятичной точке отсутствует,¹ мы можем «обмануть» `TeX` и добиться этого при помощи двух столбцов: выровненной вправо целой части и выровненной влево дробной. Команда `@{.}` в строке `\begin{tabular}` заменяет нормальный пробел между столбцами просто на «.», давая эффект одного столбца, выровненного по десятичной точке. Не забудьте заменить в ваших числах точку на разделитель столбцов (`&`)! Метку столбца можно поместить над нашим числовым «столбцом» командой `\multicolumn`:

¹Если на вашей системе установлен комплект 'tools', обратите внимание на пакет `dcolumn`.

```
\begin{tabular}{c r @{.} l}
Выражение с  $\pi$  & & \\
\multicolumn{2}{c}{Значение} & \\
\hline
 $\pi$  & 3.1416 & \\
 $\pi^{\pi}$  & 36.46 & \\
 $(\pi^{\pi})^{\pi}$  & 80662.7 & \\
\end{tabular}
```

Выражение с π	Значение
π	3.1416
π^{π}	36.46
$(\pi^{\pi})^{\pi}$	80662.7

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

Ene	
Mene	Muh!

Материал, набираемый в окружении `tabular`, всегда верстается на одной странице. Если вам нужно набирать длинные таблицы, используйте окружения `supertabular` или `longtable`.

2.12 Плавающие объекты

Большинство публикаций в наши дни содержат множество иллюстраций и таблиц. Эти элементы нуждаются в специальном обращении с ними, так как они не могут быть разбиты между страницами. Одним из выходов было бы начинать новую страницу каждый раз, когда встречается иллюстрация или таблица, слишком большая, чтобы поместиться на текущей странице. Этот подход привел бы к тому, что страницы оставались бы частично пустыми, что смотрится очень плохо.

Для решения этой проблемы любая иллюстрация или таблица, не уместящаяся на текущей странице, может ‘плавать’, перемещаясь на следующую страницу в процессе заполнения текстом текущей. $\text{\LaTeX}$ предлагает для плавающих объектов два окружения, одно для таблиц и одно для иллюстраций. Чтобы полностью использовать их преимущества, важно примерно представлять, как $\text{\LaTeX}$ обрабатывает плавающие объекты. Иначе они могут стать источником разочарования из-за того, что $\text{\LaTeX}$ помещает их не туда, куда вы хотите.

Давайте вначале рассмотрим команды, предоставляемые $\text{\LaTeX}$ для плавающих объектов.

Любой материал, включенный в окружения `figure` или `table`, трактуется как плавающий. Оба окружения имеют необязательный параметр

```
\begin{figure}[спецификация размещения] или
\begin{table}[спецификация размещения]
```

называемый *спецификацией размещения*. Этот параметр используется для указания

Л^AT_EX, куда можно перемещать плавающий объект. *Спецификация размещения* конструируется путем собирания в строчку *ключей размещения плавающего объекта*. См. таблицу 2.6.

Замечание: `0pt` и `1.05em` — единицы измерения длин T_EX. Подробности о единицах и размерностях смотрите в таблице 6.5 на странице 107.

Например, таблицу можно начать следующей строкой:

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

Спецификация размещения `[!hbp]` позволяет Л^AT_EX разместить таблицу прямо по месту (**h**), или внизу той же страницы (**b**), или на выделенной странице (**p**), и все это — даже если это будет смотреться не так уж хорошо (**!**). Если никакой спецификации размещения не задано, стандартные классы предполагают `[tbp]`.

Л^AT_EX размещает каждый встреченный плавающий объект в соответствии с заданной автором спецификацией. Если объект нельзя поместить на текущей странице, он откладывается, помещаясь в очередь иллюстраций или в очередь таблиц.¹ Когда начинается новая страница, Л^AT_EX проверяет, можно ли заполнить специальную страницу плавающими объектами из очередей. Если нет, то первый объект из каждой очереди считается только что встретившимся в тексте: Л^AT_EX снова пытается разместить их в соответствии с их спецификациями (за исключением 'h', что уже невозможно). Новые встреченные в тексте плавающие объекты помещаются в соответствующие очереди. Л^AT_EX сохраняет порядок, в котором встретились плавающие объекты соответствующего типа. Поэтому иллюстрация, которую не удастся разместить, отталкивает все дальнейшие иллюстрации к концу документа. Следовательно:

Если Л^AT_EX не размещает плавающие объекты, как вы этого ожидаете, то часто это только один объект устроил затор в одной из очередей.

¹Эти очереди подчиняются дисциплине *fifo*: 'первым вошел — первым вышел'.

Таблица 2.6: Ключи размещения плавающего объекта

Ключ	Разрешает помещать объект ...
h	<i>здесь же</i> , в том самом месте текста, где он появился. Обычно используется для маленьких объектов.
t	<i>наверху</i> страницы
b	<i>внизу</i> страницы
p	на <i>специальной странице</i> , содержащей только плавающие объекты.
!	не рассматривать большинство внутренних параметров, ^a которые могут предотвратить размещение этого объекта.

^aТаких, как максимальное число плавающих объектов, разрешенных на одной странице

Хотя и возможно задавать в $\text{\LaTeX}$ конкретную спецификацию размещения плавающего объекта, это может вызвать проблемы. Если объект не помещается в указанном месте, он «застревает», блокируя последующие плавающие объекты. В частности, никогда не используйте ключ `[h]`; это настолько плохо, что в современных версиях $\text{\LaTeX}$, он автоматически заменяется `[ht]`.

После объяснения этих механизмов остается еще несколько замечания про окружения `table` и `figure`. Командой

```
\caption{текст заголовка}
```

вы можете задать заголовок для объекта. Увеличивающийся номер и строка «Рисунок» или «Таблица» добавляются $\text{\LaTeX}$.

Две команды

```
\listoffigures и \listoftables
```

работают аналогично команде `\tableofcontents`, печатая список иллюстраций или таблиц, соответственно. В этих списках заголовки повторяются целиком. Если вы используете длинные заголовки, то вы должны предоставить их краткий вариант для включения в списки. Это делается помещением краткого варианта в квадратные скобки после команды `\caption`.

```
\caption[Короткий]{Дддддллллллииинннннныыыыыыыыыййй}
```

При помощи `\label` и `\ref` можно делать ссылки из вашего текста на плавающий объект.

Следующий пример рисует квадрат и вставляет его в документ. Подобную технику можно использовать, чтобы оставить в документе место под изображения, которые вы вставите позже.

```
1 Рисунок~\ref{white} является примером Поп-Арта.
2 \begin{figure}[!hbp]
3   \makebox[\textwidth]{\framebox[5cm]{\rule{0pt}{5cm}}}
4   \caption{Пять на пять сантиметров} \label{white}
5 \end{figure}
```

В этом примере $\text{\LaTeX}$ будет *очень сильно* (!) стараться разместить иллюстрацию прямо *по месту* (h).¹ Если это невозможно, он попытается разместить ее *внизу страницы* (b). Если ему не удастся поместить иллюстрацию на текущей странице, он выяснит, можно ли создать страницу плавающих объектов, содержащую эту иллюстрацию и, возможно, некоторые таблицы из очереди таблиц. Если для отдельной страницы материала еще не накопилось, $\text{\LaTeX}$ начинает новую страницу и снова рассматривает иллюстрацию, как если бы она только что появилась в тексте.

¹Предполагая, что очередь иллюстраций пуста

В определенных случаях может быть необходимо использовать команду

`\clearpage` или даже `\cleardoublepage`

Она указывает $\text{\LaTeX}$ немедленно разместить все плавающие объекты, остававшиеся в очередях, и затем начать новую страницу. `\cleardoublepage`, помимо этого, начинает новую правостороннюю страницу.

Позже вы узнаете, как включать в ваши документы $\text{\LaTeX}$ рисунки в формате POSTSCRIPT.

2.13 Защита хрупких команд

Текст, заданный в аргументах команд наподобие `\caption` или `\section`, появляется в документе больше одного раза (например, в оглавлении, в колонтитулах и в теле документа). Некоторые команды не работают, будучи использованы в аргументах команд типа `\section`. Их называют хрупкими командами. В частности, хрупкими являются команды `\footnote` или `\phantom`. Для того, чтобы они работали как предполагается, необходимо перед ними поставить команду `\protect`.

Команда `\protect` относится только к команде, непосредственно следующей за ней; даже не к ее аргументам. В большинстве случаев лишняя команда `\protect` не повредит.

```
\section{Я внимателен  
  \protect\footnote{и защищаю свои сноски}}
```


3 Набор математических формул

Вот теперь мы готовы! В этой главе мы встретимся с основной мощностью $\text{\TeX}$: математической версткой. Но имейте в виду, что эта глава дает только поверхностный обзор. Хотя для многих из вас изложенных здесь вещей будет достаточно, не отчаивайтесь, если вы не сможете найти решение, отвечающее нуждам верстки вашей математики. Весьма возможно, что ваша проблема решается в $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\text{\TeX}$ ¹.

3.1 Общие сведения

$\text{\TeX}$ включает в себя специальный режим для верстки математики. Математика может быть набрана внутри абзаца, но может и разбивать абзац выделенной формулой. Математический текст внутри абзаца вводится между $\backslash($ и $\backslash)$, между $\$$ и $\$$ или между $\backslash\text{begin}\{\text{math}\}$ или $\backslash\text{end}\{\text{math}\}$.

Складывая $\$a\$$ в квадрате с $\$b\$$ в квадрате, получаем $\$c\$$ в квадрате. Или излагая языком математики: $\$c^{\text{\texttt{2}}}=a^{\text{\texttt{2}}}+b^{\text{\texttt{2}}}\$$

Складывая a в квадрате с b в квадрате, получаем c в квадрате. Или излагая языком математики: $c^2 = a^2 + b^2$

$\backslash\text{\texttt{TeX}}\{\}$ произносится как $\backslash(\backslash\text{\texttt{tau}}\backslash\text{\texttt{epsilon}}\backslash\text{\texttt{chi}}\backslash).\backslash\backslash[6\text{\texttt{pt}}]$
 $100\sim\text{\texttt{m}}\text{\texttt{f}}^{\text{\texttt{3}}}\$$ воды. $\backslash\backslash[6\text{\texttt{pt}}]$
Это исходит от моего $\backslash\text{begin}\{\text{math}\}\backslash\text{heartsuit}\backslash\text{end}\{\text{math}\}$

$\text{\TeX}$ произносится как $\tau\epsilon\chi$.
 100 м^3 воды.
Это исходит от моего ♡

Большие математические уравнения или формулы предпочтительнее «выключать», то есть верстать их на отдельных строках. Для этого заключайте их между $\backslash[$ и $\backslash]$ или между $\backslash\text{begin}\{\text{displaymath}\}$ и $\backslash\text{end}\{\text{displaymath}\}$.

¹ *Американское Математическое Общество* выработало развитое расширение к $\text{\TeX}$. Многие примеры этой главы используют это расширение, входящее во все современные дистрибутивы $\text{\TeX}$. Если в вашем оно отсутствует, то можете получить его по адресу `macros/latex/required/amslatex`.

Складывая a в квадрате с b в квадрате, получаем c в квадрате. Или излагая языком математики:

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

или вы можете выразить это короче: $a+b=c$

Складывая a в квадрате с b в квадрате, получаем c в квадрате. Или излагая языком математики:

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

или вы можете выразить это короче:

$$a + b = c$$

Если вы хотите, чтобы $\text{\LaTeX}$ нумеровал ваши уравнения, используйте окружение `equation`. Вы можете при этом пометить уравнение меткой `\label` и сослаться на него в любом месте текста командами `\ref` или `\eqref`:

```
\begin{equation}
\label{eq:eps}
\epsilon > 0
\end{equation}
Из (\ref{eq:eps})
выводим \ldots{И}з
\eqref{eq:eps} мы
делаем то же.
```

$$\epsilon > 0 \tag{3.1}$$

Из (3.1) выводим ... Из (3.1) мы делаем то же.

Заметьте разницу в стиле верстки выражений в абзацах и выключных:

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

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

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

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

Математический режим отличается от *текстового режима*. Например, в *математическом режиме*:

1. Большинство пробелов и возвратов каретки не принимаются во внимание, так как все пробелы либо выводятся из логики математических выражений, или должны в явном виде задаваться командами вроде `\,`, `\quad` или `\qquad`.
2. Пустые строчки недопустимы. Каждая формула занимает только один абзац.
3. Каждая буква считается именем переменной, и верстается в этом качестве. Если вы хотите в формулу ввести нормальный текст (нормальный прямой шрифт с нормальными пробелами), то вам нужно вводить его командами `\textrm{...}` (см. также раздел 3.7 на стр. 56).

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

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

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

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

Математики бывают очень строги к используемым символам: здесь будет удобно использовать ‘ажурные полужирные символы’, которые получаются командой `\mathbb` из пакетов `amsmath` или `amssymb`. Последний пример теперь выглядит так:

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

$$x^2 \geq 0 \quad \text{для всех } x \in \mathbb{R}$$

3.2 Группировка в математическом режиме

Большинство команд математического режима действует только на следующий символ. Так что, если вы хотите, чтобы команда влияла на несколько символов, вам нужно сгруппировать их вместе при помощи фигурных скобок: `\{...\}`.

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

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

3.3 Составляющие математической формулы

В этом разделе будут описаны наиболее важные команды, используемые в математической верстке. Детальный перечень команд для набора математических символов смотрите в разделе 3.10 на странице 60.

Строчные греческие буквы вводятся как `\alpha`, `\beta`, `\gamma`, ..., прописные буквы вводятся как `\Gamma`, `\Delta`, ...¹

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

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

¹ В $\text{\LaTeX}$ 2_ε не определяется прописная «альфа», потому что она выглядит так же, как латинская «А». При новой кодировке математики это будет изменено.

Верхние и нижние индексы вводятся при помощи символов «^» и «_».

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

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

Квадратный корень вводится как `\sqrt`, корень n -ной степени печатается при помощи `\sqrt[n]`. Размер знака корня выбирается L^AT_EX автоматически. Если нужен один только знак, используйте `\surd`.

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

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

Команды `\overline` и `\underline` создают **горизонтальные линии** сразу над или под выражением.

```
$\overline{m+n}$
```

$$\overline{m+n}$$

Команды `\overbrace` и `\underbrace` создают длинные **горизонтальные фигурные скобки** сразу над или под выражением.

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

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

Для добавления к переменным знаков математических акцентов, таких, как маленькие стрелки или знака тильда, вы можете пользоваться командами, перечисленным в таблице 3.1. Широкие «шляпки» и тильды, охватывающие несколько символов, генерируются командами `\widetilde` и `\widehat`. Символ «'» дает знак производной.

```
\begin{displaymath}
y=x^2 \quad y'=2x \\
\quad y''=2 \\
\end{displaymath}
```

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

Векторы часто указываются добавлением маленьких стрелок стрелки над переменной. Это делается командой `\vec`. Для обозначения вектора от A до B полезны две команды `\overrightarrow` и `\overleftarrow`.

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

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

Обычно знак точки, обозначающий умножения, явно не набирается. Однако, иногда он полезен, чтобы помочь читателю сгруппировать формулу. Используйте для этого `\cdot`:

```
\begin{displaymath}
v = {\sigma}_1 \cdot {\sigma}_2
{\tau}_1 \cdot {\tau}_2
\end{displaymath}
```

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

Имена функций типа `lg` часто верстаются прямым шрифтом, а не курсивом, как переменные. Поэтому $\text{\LaTeX}$ содержит следующие команды для набора имен наиболее важных функций:

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

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

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

Для функции модуля функция модуля есть две команды: `\bmod` для бинарного оператора « $a \bmod b$ » и `\pmod` для выражений вроде « $x \equiv a \pmod{b}$ ».

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

$$a \bmod b$$

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

Двухъязысная **дробь** верстается командой `\frac{\dots}{\dots}`. Часто предпочтительнее ее форма с косой чертой $1/2$, потому что она смотрится лучше при небольшом количестве ‘дробного материала’.

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

$$1\frac{1}{2} \text{ часа}$$

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

Для верстки биномиальных коэффициентов или аналогичных структур можно пользоваться командой `\binom` из пакета `amsmath`.

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

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

Для бинарных отношений бывает удобно размещать символы друг над другом. Команда `\stackrel` набирает символ, заданный первым аргументом, шрифтом размера индексов и размещает его над вторым аргументом, верстаемым в обычной позиции:

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

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

Оператор интеграла печатает команда `\int`, **оператор суммы** — команда `\sum`, **оператор произведения** — команда `\prod`. Верхние и нижние пределы указываются при помощи знаков «^» и «_», так же, как верхние и нижние индексы¹.

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

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

Чтобы получить больший контроль над размещением индексов в сложных выражениях, `amsmath` предоставляет еще два инструмента: команду `\substack` и окружение `subarray`:

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

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

Для **скобок** и прочих ограничителей в `TeX` существует множество символов (скажем, `[` `<` `||` `\updownarrow`). Круглые и квадратные скобки можно вводить соответствующими клавишами, фигурные скобки — `\{`, прочие ограничители — специальными командами (например, `\updownarrow`). Список доступных ограничителей смотрите в таблице 3.8 на странице 62.

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

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

Если вы поместите перед открывающим ограничителем команду `\left`, или перед закрывающим — `\right`, то `TeX` автоматически выберет правильный размер ограничителя. Заметьте, что вы должны каждый `\left` закрывать соответствующим `\right`, и что размер определяется корректно только если оба они набраны на

¹ `AMS-TeX`, кроме того, поддерживает многострочные верхние и нижние индексы.

одной строке. Если вы не хотите иметь правого ограничителя, используйте невидимый ограничитель ‘\right.’!

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

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

В некоторых случаях необходимо указать корректный размер математического ограничителя вручную, для этого есть команды `\big`, `\Big`, `\bigg` и `\Bigg`, служащие префиксами к большинству команд ограничителей.¹

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

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

$$\left(\left(\left(\left(\right) \right) \right) \right) \quad \left\| \left\| \left\| \left\| \right\| \right\| \right\|$$

Чтобы ввести в формулу **три точки**, есть несколько команд. `\ldots` верстаёт точки на базовой линии, `\cdots` — центрированные. Кроме того, существуют команды `\vdots` для вертикальных и `\ddots` для диагональных точек. В разделе 3.5 вы найдёте другой пример.

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

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

3.4 Математические пробелы

Если выбранные $\TeX$ пробелы внутри формул неудовлетворительны, вы можете их подстраивать с использованием команд управления пробелами. Команды для маленьких пробелов: `\`, для $\frac{3}{18}$ quad (U), `\:` для $\frac{4}{18}$ quad (U) и `\;` для $\frac{5}{18}$ quad (U). Экранированный символ пробела `\` даёт средних размеров пробел, а `\quad` (L) и `\qquad` (L) дают большие пробелы. Размер `\quad` примерно соответствует ширине буквы ‘M’ в текущем шрифте. Команда `\!` производит отрицательный пробел размером $-\frac{3}{18}$ quad (U).

¹Эти команды не работают как ожидается, если используются команды смены размера шрифта, или если указана опция `11pt` или `12pt`. Для корректирования этого поведения пользуйтесь пакетами `exscale` или `amsmath`.

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

вместо

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

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

Заметьте, что ‘d’ в дифференциале обычно верстается прямым шрифтом.

$\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{I}\mathcal{T}\mathcal{E}\mathcal{X}$ включает другой способ тонкой подстройки пробелов между несколькими знаками интегралов: команды `\iint`, `\iiint`, `\iiint` и `\idotsint`. С загруженным пакетом `amsmath` предыдущий пример можно набирать так:

```
\newcommand{\ud}{\mathrm{d}}
\begin{displaymath}
\iint\limits_D \, \, \, \mathrm{d}x \, \, \, \mathrm{d}y
\end{displaymath}
```

$$\iint_D dx \, dy$$

Детали смотрите в электронном документе `textmath.tex` (распространяется с $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{I}\mathcal{T}\mathcal{E}\mathcal{X}$) или в главе 8 *The $\mathcal{I}\mathcal{T}\mathcal{E}\mathcal{X}$ Companion* [3].

3.5 Вертикально расположенный материал

Для верстки **матриц** пользуйтесь окружением `array`. Его работа напоминает окружение `tabular`. Для разрыва строки используется команда `\\`.

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

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

Окружение `array` можно также использовать для верстки выражений, имеющих один большой ограничитель, подставляя «.» в качестве невидимого правого ограничителя:

```
\begin{displaymath}
y = \left\{ \begin{array}{ll}
a & \text{если } d > c \\
b+x & \text{по утрам} \\
l & \text{остальное время дня}
\end{array} \right.
\end{displaymath}
```

$$y = \begin{cases} a & \text{если } d > c \\ b+x & \text{по утрам} \\ l & \text{остальное время дня} \end{cases}$$

Так же, как в окружении `tabular`, можно рисовать линейки в окружении `array`, например, разделяя элементы матрицы:

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

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

Для формул, занимающих несколько строк или для систем уравнений вместо `equation` пользуйтесь окружениями `eqnarray` и `eqnarray*`. В `eqnarray` каждая строка получает отдельный номер уравнения. В `eqnarray*` номера не ставятся.

Окружения `eqnarray` и `eqnarray*` работают наподобие таблицы из трех столбцов формата `{rc1}`, где средний столбец используется для знака равенства, или знака неравенства, или другого подходящего знака. Команда `\\` разбивает строки.

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

$$\begin{aligned} f(x) &= \cos x & (3.5) \\ f'(x) &= -\sin x & (3.6) \\ \int_0^x f(y)dy &= \sin x & (3.7) \end{aligned}$$

Заметьте, что по обеим сторонам средней колонки, знаков равенства, слишком много свободного места. Оно может быть уменьшено установкой `\setlength\arraycolsep{2pt}`, как в следующем примере.

Длинные уравнения не будут автоматически разбиваться на правильные части. Автор должен указать, где их разбивать и насколько выравнивать. Чаще всего для этого используют следующие методы:

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

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

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

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

Команда `\nonumber` заставляет $\text{\LaTeX}$ не генерировать номер для этого уравнения.

Такими методами может быть сложно получить правильно выглядящие вертикально выровненные уравнения; более мощную альтернативу предоставляет пакет `amsmath` (см. окружения `align`, `flalign`, `gather`, `multiline` и `split`).

3.6 Фантомы

Мы не можем увидеть привидения, но они, тем не менее, все равно занимают свое место в умах некоторых людей. Так и $\text{\LaTeX}$ позволяет использовать фантомы, невидимые объекты, для реализации интересных трюков с размещением видимых объектов.

Когда $\text{\LaTeX}$ размещает текст по вертикали при помощи команд `^` и `_`, он иногда проявляет многовато интеллекта. Командой `\phantom` вы можете зарезервировать пространство для символов, которых на самом деле сверстано не будет. Лучше всего это понять на следующих примерах.

```
\begin{displaymath}
{}^{12}_{\phantom{1}6}\text{\texttrm{C}}
\quad \quad \quad \backslashqquad \backslashtexttrm{versus} \quad \quad \quad \backslashqquad
{}^{12}_{6}\text{\texttrm{C}}
\end{displaymath}
```

$${}^{12}_{6}\text{C} \quad \text{versus} \quad {}^{12}_6\text{C}$$

```
\begin{displaymath}
\Gamma_{ij}^{\phantom{ij}k}
\quad \quad \quad \backslashqquad \backslashtexttrm{versus} \quad \quad \quad \backslashqquad
\Gamma_{ij}^k
\end{displaymath}
```

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

3.7 Размер математического шрифта

В математическом режиме $\text{\TeX}$ выбирает размер шрифта в зависимости от контекста. Индексы, например, верстаются меньшим шрифтом. Если вы хотите добавить к уравнению обычный текст, не пользуйтесь командой `\texttrm`, так как механизм переключения размера работать не будет, потому что `\texttrm` временно выходит в

Аргументы в квадратных скобках необязательны. Оба они используются для определения того, как нумеровать «теорему». Аргументом *счетчик* вы можете указать *название* предварительно объявленной «теоремы». Новая «теорема» будет тогда нумероваться в той же последовательности. Аргумент *раздел* позволяет вам указать раздел, внутри которого вы хотите нумеровать вашу «теорему».

После использования в преамбуле документа команды `\newtheorem`, вы можете пользоваться следующими командами:

```
\begin{название}[текст]
Это интересная теорема.
\end{название}
```

На этом теории должно быть достаточно. Дальнейшие примеры должны развеять последнюю тень сомнений, и окончательно убедить вас, что окружение `\newtheorem` слишком сложно, чтобы его можно было понять:

```
% определения для
% преамбулы документа
\newtheorem{law}{Law}
\newtheorem{jury}{law}[law]{Jury}
% в теле документа
\begin{law} \label{law:box}
Don't hide in the witness box
\end{law}
\begin{jury}[The Twelve]
It could be you! So beware and
see law~\ref{law:box}\end{jury}
\begin{law}No, No, No\end{law}
```

Law 1 *Don't hide in the witness box*

Jury 2 (The Twelve) *It could be you! So beware and see law 1*

Law 3 *No, No, No*

Теорема «Jury» использует тот же счетчик, что и теорема «Law». Следовательно, она получит номер в последовательности с другими теоремами «Law». Аргумент в квадратных скобках указывает заголовок теоремы, или нечто аналогичное.

```
\flushleft
\newtheorem{mur}{Murphy}[section]
\begin{mur}
Если существует два или
более способа сделать
нечто, и один из этих
способов может привести
к катастрофе, то кто-то
обязательно это сделает.
\end{mur}
```

Murphy 3.8.1 *Если существует два или более способа сделать нечто, и один из этих способов может привести к катастрофе, то кто-то обязательно это сделает.*

Теорема «Murphy» получает номер, связанный с номером текущего раздела. Вы можете также использовать другую структурную единицу, например, главу или подраздел.

3.9 Полужирные символы

В $\text{\LaTeX}$ довольно непросто получить жирные символы; это, вероятно, сделано преднамеренно, потому что непрофессионалы слишком часто злоупотребляют ими. Команда смены шрифта `\mathbf` дает полужирные символы, но они обычные (прямые), тогда как математические символы обычно курсивные. Существует команда `\boldmath`, но *она может использоваться только вне математического режима*. То же относится и к символам.

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

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

Заметьте, что запятая тоже полужирная, что может быть нежелательным.

Пакет `amsbsy` (включаемый пакетом `amsmath`), равно как и пакет `bm` (из набора `tools`), включает команду `\boldsymbol`.

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

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

3.10 Список математических символов

В следующих таблицах вы найдете все символы, известные обычно в *математическом режиме*.

Для доступа к символам, перечисленным в таблицах 3.12–3.16¹ в преамбуле документа должен быть загружен пакет `amssymb`, и в системе должны быть установлены математические шрифты AMS. Если пакеты и шрифты AMS в вашей системе не установлены, посмотрите на `CTAN:/tex-archive/macros/latex/required/amslatex`. Еще более полный перечень символов можно найти по адресу `info/symbols/comprehensive`.

Таблица 3.1: Акценты математического режима

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

Таблица 3.2: Строчные греческие буквы

α	<code>\alpha</code>	θ	<code>\theta</code>	o	<code>o</code>	v	<code>\upsilon</code>
β	<code>\beta</code>	ϑ	<code>\vartheta</code>	π	<code>\pi</code>	ϕ	<code>\phi</code>
γ	<code>\gamma</code>	ι	<code>\iota</code>	ϖ	<code>\varpi</code>	φ	<code>\varphi</code>
δ	<code>\delta</code>	κ	<code>\kappa</code>	ρ	<code>\rho</code>	χ	<code>\chi</code>
ϵ	<code>\epsilon</code>	λ	<code>\lambda</code>	ϱ	<code>\varrho</code>	ψ	<code>\psi</code>
ε	<code>\varepsilon</code>	μ	<code>\mu</code>	σ	<code>\sigma</code>	ω	<code>\omega</code>
ζ	<code>\zeta</code>	ν	<code>\nu</code>	ς	<code>\varsigma</code>		
η	<code>\eta</code>	ξ	<code>\xi</code>	τ	<code>\tau</code>		

Таблица 3.3: Прописные греческие буквы

Γ	<code>\Gamma</code>	Λ	<code>\Lambda</code>	Σ	<code>\Sigma</code>	Ψ	<code>\Psi</code>
Δ	<code>\Delta</code>	Ξ	<code>\Xi</code>	Υ	<code>\Upsilon</code>	Ω	<code>\Omega</code>
Θ	<code>\Theta</code>	Π	<code>\Pi</code>	Φ	<code>\Phi</code>		

¹Эти таблицы были получены из `symbols.tex` (автор David Carlisle) и позже сильно изменены по совету Josef Tkadlec

Таблица 3.4: Бинарные отношения

Вы можете получить соответствующие отрицания добавлением перед следующими символами команды `\not`.

$<$	<code><</code>	$>$	<code>></code>	$=$	<code>=</code>
$\leq$	<code>\leq</code> или <code>\le</code>	$\geq$	<code>\geq</code> или <code>\ge</code>	$\equiv$	<code>\equiv</code>
$\ll$	<code>\ll</code>	$\gg$	<code>\gg</code>	$\doteq$	<code>\doteq</code>
$\prec$	<code>\prec</code>	$\succ$	<code>\succ</code>	$\sim$	<code>\sim</code>
$\preceq$	<code>\preceq</code>	$\succeq$	<code>\succeq</code>	$\simeq$	<code>\simeq</code>
$\subset$	<code>\subset</code>	$\supset$	<code>\supset</code>	$\approx$	<code>\approx</code>
$\subseteq$	<code>\subseteq</code>	$\supseteq$	<code>\supseteq</code>	$\cong$	<code>\cong</code>
$\sqsubset$	<code>\sqsubset</code> ¹	$\sqsupset$	<code>\sqsupset</code> ¹	$\Join$	<code>\Join</code> ¹
$\sqsubseteq$	<code>\sqsubseteq</code>	$\sqsupseteq$	<code>\sqsupseteq</code>	$\bowtie$	<code>\bowtie</code>
$\in$	<code>\in</code>	$\ni$	<code>\ni</code> , <code>\owns</code>	$\propto$	<code>\propto</code>
$\vdash$	<code>\vdash</code>	$\dashv$	<code>\dashv</code>	$\models$	<code>\models</code>
$\mid$	<code>\mid</code>	$\parallel$	<code>\parallel</code>	$\perp$	<code>\perp</code>
$\smile$	<code>\smile</code>	$\frown$	<code>\frown</code>	$\asymp$	<code>\asymp</code>
$:$	<code>:</code>	$\notin$	<code>\notin</code>	$\neq$	<code>\neq</code> или <code>\ne</code>

¹Для доступа к этому символу пользуйтесь пакетом `latexsym`.

Таблица 3.5: Бинарные операторы

$+$	<code>+</code>	$-$	<code>-</code>		
$\pm$	<code>\pm</code>	$\mp$	<code>\mp</code>	$\triangleleft$	<code>\triangleleft</code>
$\cdot$	<code>\cdot</code>	$\div$	<code>\div</code>	$\triangleright$	<code>\triangleright</code>
$\times$	<code>\times</code>	$\setminus$	<code>\setminus</code>	$\star$	<code>\star</code>
$\cup$	<code>\cup</code>	$\cap$	<code>\cap</code>	$\ast$	<code>\ast</code>
$\sqcup$	<code>\sqcup</code>	$\sqcap$	<code>\sqcap</code>	$\circ$	<code>\circ</code>
$\vee$	<code>\vee</code> , <code>\lor</code>	$\wedge$	<code>\wedge</code> , <code>\land</code>	$\bullet$	<code>\bullet</code>
$\oplus$	<code>\oplus</code>	$\ominus$	<code>\ominus</code>	$\diamond$	<code>\diamond</code>
$\odot$	<code>\odot</code>	$\oslash$	<code>\oslash</code>	$\uplus$	<code>\uplus</code>
$\otimes$	<code>\otimes</code>	$\bigcirc$	<code>\bigcirc</code>	$\amalg$	<code>\amalg</code>
$\triangle$	<code>\bigtriangleup</code>	∇	<code>\bigtriangledown</code>	$\dagger$	<code>\dagger</code>
$\triangleleft$	<code>\lhd</code> ¹	$\triangleright$	<code>\rhd</code> ¹	$\ddagger$	<code>\ddagger</code>
$\trianglelefteq$	<code>\unlhd</code> ¹	$\trianglerighteq$	<code>\unrhd</code> ¹	$\wr$	<code>\wr</code>

Таблица 3.6: Большие операторы

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

Таблица 3.7: Стрелки

$\leftarrow$	<code>\leftarrow</code> или <code>\gets</code>	$\longleftarrow$	<code>\longleftarrow</code>	$\uparrow$	<code>\uparrow</code>
$\rightarrow$	<code>\rightarrow</code> или <code>\to</code>	$\longrightarrow$	<code>\longrightarrow</code>	$\downarrow$	<code>\downarrow</code>
$\leftrightarrow$	<code>\leftrightarrow</code>	$\longleftrightarrow$	<code>\longleftrightarrow</code>	$\updownarrow$	<code>\updownarrow</code>
$\Leftarrow$	<code>\Leftarrow</code>	$\Longleftarrow$	<code>\Longleftarrow</code>	$\Uparrow$	<code>\Uparrow</code>
$\Rightarrow$	<code>\Rightarrow</code>	$\Longrightarrow$	<code>\Longrightarrow</code>	$\Downarrow$	<code>\Downarrow</code>
$\Leftrightarrow$	<code>\Leftrightarrow</code>	$\Longleftrightarrow$	<code>\Longleftrightarrow</code>	$\Updownarrow$	<code>\Updownarrow</code>
$\mapsto$	<code>\mapsto</code>	$\longmapsto$	<code>\longmapsto</code>	$\nearrow$	<code>\nearrow</code>
$\hookrightarrow$	<code>\hookrightarrow</code>	$\hookrightarrow$	<code>\hookrightarrow</code>	$\searrow$	<code>\searrow</code>
$\leftharpoonup$	<code>\leftharpoonup</code>	$\rightharpoonup$	<code>\rightharpoonup</code>	$\swarrow$	<code>\swarrow</code>
$\leftharpoondown$	<code>\leftharpoondown</code>	$\rightharpoondown$	<code>\rightharpoondown</code>	$\nwarrow$	<code>\nwarrow</code>
$\rightleftharpoons$	<code>\rightleftharpoons</code>	$\iff$ (бóльший пробел)	<code>\iff</code> (бóльший пробел)	$\leadsto$	<code>\leadsto</code> ¹

¹Для доступа к этому символу пользуйтесь пакетом `latexsym`.

Таблица 3.8: Ограничители

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

Таблица 3.9: Большие ограничители

$\left($	<code>\lgrou</code>	$\right)$	<code>\rgrou</code>	$\left[$	<code>\lmoustache</code>	$\right]$	<code>\rmoustache</code>
$\uparrow$	<code>\arrowvert</code>	$\Uparrow$	<code>\Arrowvert</code>	$\lceil$	<code>\bracevert</code>		

Таблица 3.10: Прочие символы

...	<code>\dots</code>	...	<code>\cdots</code>	:	<code>\vdots</code>	⋯	<code>\ddots</code>
$\hbar$	<code>\hbar</code>	$\imath$	<code>\imath</code>	$\jmath$	<code>\jmath</code>	ℓ	<code>\ell</code>
$\Re$	<code>\Re</code>	$\Im$	<code>\Im</code>	$\aleph$	<code>\aleph</code>	$\wp$	<code>\wp</code>
$\forall$	<code>\forall</code>	$\exists$	<code>\exists</code>	$\mho$ ¹	<code>\mho</code>	∂	<code>\partial</code>
'	<code>'</code>	'	<code>\prime</code>	$\emptyset$	<code>\emptyset</code>	∞	<code>\infty</code>
∇	<code>\nabla</code>	$\triangle$	<code>\triangle</code>	$\Box$ ¹	<code>\Box</code>	$\diamond$ ¹	<code>\Diamond</code>
$\bot$	<code>\bot</code>	$\top$	<code>\top</code>	$\angle$	<code>\angle</code>	$\surd$	<code>\surd</code>
$\diamondsuit$	<code>\diamondsuit</code>	$\heartsuit$	<code>\heartsuit</code>	$\clubsuit$	<code>\clubsuit</code>	$\spadesuit$	<code>\spadesuit</code>
$\neg$	<code>\neg</code> или <code>\lnot</code>	$\flat$	<code>\flat</code>	$\natural$	<code>\natural</code>	$\sharp$	<code>\sharp</code>

¹Для доступа к этому символу пользуйтесь пакетом `latexsym`.

Таблица 3.11: Не-математические символы

Эти символы можно использовать и в текстовом режиме.

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

Таблица 3.12: Ограничители AMS

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

Таблица 3.13: Буквы греческого и иврита AMS

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

Таблица 3.14: Бинарные отношения AMS

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

Таблица 3.15: Стрелки AMS

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

Таблица 3.16: Отрицательные бинарные отношения и стрелки AMS

$\nless$	$\ngtr$	$\varsubsetneqq$
$\lneq$	$\gneq$	$\varsupsetneqq$
$\nleq$	$\ngeq$	$\nsubseteqeq$
$\nleqslant$	$\ngeqslant$	$\nsupseteqeq$
$\lneqq$	$\gneqq$	$\nmid$
$\lvertneqq$	$\gvertneqq$	$\nparallel$
$\nleqq$	$\ngeqq$	$\nshortmid$
$\lnsim$	$\gnsim$	$\nshortparallel$
$\lnapprox$	$\gnapprox$	$\nsim$
$\nprec$	$\nsucc$	$\ncong$
$\npreceq$	$\nsucceq$	$\nvdash$
$\precneqq$	$\succneqq$	$\nvDash$
$\precnsim$	$\succnsim$	$\nVdash$
$\precnapprox$	$\succnapprox$	$\nVDash$
$\subsetneq$	$\supsetneq$	$\ntriangleleft$
$\varsubsetneq$	$\varsupsetneq$	$\ntriangleright$
$\nsubseteq$	$\nsupseteq$	$\ntrianglelefteq$
$\subsetneqq$	$\supsetneqq$	$\ntrianglerighteq$
$\nleftarrow$	$\rightarrow$	$\nleftrightarrow$
$\nLeftarrow$	$\nrightarrow$	$\nLeftrightarrow$

Таблица 3.17: Бинарные операторы AMS

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

Таблица 3.18: Прочие символы AMS

$\hbar$	<code>\hbar</code>	$\hslash$	<code>\hslash</code>	$\Bbbk$	<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>
$\lozenge$	<code>\lozenge</code>	$\blacklozenge$	<code>\blacklozenge</code>	$\bigstar$	<code>\bigstar</code>
$\angle$	<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>		

Таблица 3.19: Математические алфавиты

Пример	Команда	Требуемый пакет
ABCdef	<code>\mathrm{ABCdef}</code>	
ABCdef	<code>\mathit{ABCdef}</code>	
$\mathnormal{ABCdef}$	<code>\mathnormal{ABCdef}</code>	
$\mathcal{ABC}$	<code>\mathcal{ABC}</code>	euscript с опцией: <code>mathcal</code>
$\mathscr{ABC}$	<code>\mathscr{ABC}</code>	<code>mathrsfs</code>
$\mathfrak{ABCdef}$	<code>\mathfrak{ABCdef}</code>	<code>eufrak</code>
$\mathbb{ABC}$	<code>\mathbb{ABC}</code>	<code>amssymb</code> или <code>amssymb</code>

4 Специальные возможности

При сборке большого документа $\text{\LaTeX}$ поможет вам некоторыми специальными функциями, например, генерацией предметного указателя, управлением библиографией и тому подобными. Более подробное описание специальных возможностей и расширений $\text{\LaTeX}$ находится в *$\text{\LaTeX}$ Manual* [1] и в *The $\text{\LaTeX}$ Companion* [3].

4.1 Включение Encapsulated PostScript графики

$\text{\LaTeX}$ имеет базовые средства для работы с плавающими объектами, такими, как иллюстрации и таблицы, при помощи окружений **figure** и **table**.

Существует также несколько способов создавать собственно графику средствами базового $\text{\LaTeX}$ или его расширений. К сожалению, большинство пользователей находит их трудными для понимания, поэтому здесь эти способы не рассматриваются. Дополнительная информация приведена в *The $\text{\LaTeX}$ Companion* [3] и в *$\text{\LaTeX}$ Manual* [1].

Более простой метод получения графики в документе — это создавать ее специализированными программными пакетами¹ и включать в документ готовую графику. Пакеты $\text{\LaTeX}$ предлагают множество способов это делать. В этом введении обсуждается только использование графики в формате Encapsulated PostScript (EPS), поскольку это довольно просто делается и широко распространено. Чтобы использовать картинки в формате EPS, вам нужно использовать для вывода POSTSCRIPT принтер.²

Хороший набор команд включения графики входит в пакет **graphicx** (автор D. P. Carlisle). Он является частью целого семейства пакетов, называющегося комплектом “graphics”.³

В предположении, что вы работаете в системе с доступным для вывода POSTSCRIPT-принтером, и с установленным пакетом **graphicx**, для включения картинки в ваш документ можете использовать следующую пошаговую инструкцию:

1. Экспортируйте картинку из вашей графической программы в формате EPS.⁴

¹Такими, как XFig, CorelDraw!, Freehand, Gnuplot, ...

²Другая возможность вывода POSTSCRIPT заключается в использовании программы GHOSTSCRIPT, доступной с `CTAN:/tex-archive/support/ghostscript`. Пользователи Windows и OS/2 могут обратить внимание на программу GSVIEW.

³`macros/latex/required/graphics`

⁴Если ваша программа не имеет экспорта в EPS, можно попробовать установить драйвер

2. В преамбуле документа загрузите пакет `graphicx` при помощи

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

где *драйвер* — это название вашего конвертера DVI в PostScript. Самый широко используемый называется `dvips`. Название драйвера требуется потому, что не существует стандарта включения графики в TeX. Зная название *драйвера*, `graphicx` может выбрать правильный способ вставить информацию о графике в `.dvi` файл так, чтобы драйвер ее понял и смог корректно вставить `.eps` файл.

3. Для включения *файла* в ваш документ пользуйтесь командой

```
\includegraphics[опция=значение, ...]{файл}
```

Необязательный параметр принимает вид списка разделенных запятыми пар *опций* и соответствующих *значений*. *Опции* можно использовать для изменения ширины, высоты, поворота включаемой графики. Таблица 4.1 перечисляет самые важные опции.

Таблица 4.1: Названия опций пакета `graphicx`

<code>width</code>	масштабирует графику до указанной ширины
<code>height</code>	масштабирует графику до указанной высоты
<code>angle</code>	поворачивает графику против часовой стрелки
<code>scale</code>	масштабирует графику

Возможно, пример сделает это понятнее:

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

Здесь включается графика, записанная в файл `test.eps`. Она *сначала* поворачивается на 90 градусов и *затем* масштабируется до конечной ширины в 0.5 ширины стандартного абзаца. Пропорции сохраняются, поскольку не указано конкретной

PostScript-принтера (скажем, какого-нибудь Apple LaserWriter) и печатать в файл через этот драйвер. Если вам повезет, в этом файле будет EPS. Заметьте, что EPS обязан содержать не больше одной страницы. Некоторые принтерные драйверы нужно в явном виде настраивать для генерации EPS.

высоты. Параметры высоты и ширины могут также быть указаны в абсолютных размерах. Обратитесь к таблице 6.5 на странице 107. Если вы хотите знать об этом больше, прочтите [9] и [13].

4.2 Библиография

Окружение `thebibliography` генерирует библиографию. Каждый элемент начинается с

```
\bibitem[метка]{маркер}
```

Затем *маркер* используется для дальнейших ссылок на книгу, статью или труд.

```
\cite{маркер}
```

Если вы не используете опцию *метка*, элементы библиографии нумеруются автоматически. Параметр после команды `\begin{thebibliography}` устанавливает максимальную ширину этих номеров. В следующем примере `{99}` указывает $\text{\LaTeX}$, что ни один из номеров элементов не будет шире, чем число ‘99’.

```
Partl~\cite{pa} предложил,  
чтобы \ldots  
  
\begin{thebibliography}{99}  
\bibitem{pa} Н.~Partl:  
\emph{German \TeX},  
TUGboat Volume~9, Issue~1 ('88)  
\end{thebibliography}
```

Partl [1] предложил, чтобы ...

Литература

[1] Н. Partl: *German $\text{\TeX}$* , TUGboat Volume 9, Issue 1 ('88)

Для использования в больших проектах обратите внимание на программу Bib $\text{\TeX}$. Она входит в большинство поставок $\text{\TeX}$. Bib $\text{\TeX}$ позволяет вам поддерживать библиографическую базу данных и извлекать из нее ссылки, имеющие отношение к тому, что вы цитировали в вашем труде. Визуальное представление библиографий, генерируемых Bib $\text{\TeX}$, основано на концепции стилей, что позволяет вам создавать библиографии, следуя любому из существующих стилей библиографии.

Таблица 4.2: Примеры синтаксиса ключей указателя

Пример	Вид указателя	Комментарий
<code>\index{hello}</code>	hello, 1	Обычный элемент
<code>\index{hello!Peter}</code>	Peter, 3	Подчиненный ‘hello’ элемент
<code>\index{Sam@\textsl{Sam}}</code>	Sam, 2	Форматированный ключ
<code>\index{Lin@\textbf{Lin}}</code>	Lin , 7	То же
<code>\index{Jenny textbf}</code>	Jenny, 3	Форматированная страница
<code>\index{Joe textit}</code>	Joe, 5	То же
<code>\index{eolienne@\'eolienne}</code>	éolienne, 4	Акцентированные буквы

4.3 Указатели

Полезной особенностью многих книг является их предметный указатель. Указатели могут создаваться автоматически при помощи $\text{\LaTeX}$ и сопровождающей программы `makeindex`.¹ В этом введении будут изложены только базовые команды генерации указателя. Более глубокое изложение смотрите в *The $\text{\LaTeX}$ Companion* [3].

Для включения возможностей $\text{\LaTeX}$ в преамбуле должен загружаться пакет `makeidx`:

```
\usepackage{makeidx}
```

а специальные команды должны быть разрешены помещением в преамбулу команды

```
\makeindex
```

Содержимое указателя создается командами

```
\index{ключ}
```

где *ключ* является элементом указателя. Вы вводите команды указателя в том месте текста, куда этот элемент должен указывать. Таблица 4.2 объясняет синтаксис аргумента *ключ* несколькими примерами.

По мере обработки входного файла $\text{\LaTeX}$, каждая команда `\index` записывает соответствующий элемент указателя вместе с номером текущей страницы в специальный файл. Файл имеет то же имя, что и входной файл $\text{\LaTeX}$, но другое расширение

¹На системах, не поддерживающих длинные имена файлов, программа может называться `makeidx`.

имени (`.idx`). Этот `.ind`-файл затем обрабатывается программой `makeindex`.

`makeindex filename`

Программа `makeindex` генерирует отсортированный указатель с тем же именем, но, на этот раз, с расширением `.ind`.

Если теперь повторно обработать входной файл, этот отсортированный указатель включается в документ в то место, где $\text{\LaTeX}$ находит команду

`\printindex`

Пакет `showidx`, входящий в $\text{\LaTeX}$ 2_ε, печатает все элементы указателя на левом поле текста. Это весьма полезно при проверке текста и сверке указателя.

Заметим, что команда `\index`, будучи использована неосторожно, может повлиять на вид верстки.

Мое Слово `\index{Слово}`. В отличие от `Слово\index{Слово}`. Отметьте положение точки.

Мое Слово . В отличие от Слово. Отметьте положение точки.

4.4 Настраиваемые колонтитулы

Пакет `fancyhdr`,¹ написанный Piet van Oostrum, предоставляет несколько простых команд, позволяющих вам настраивать верхние и нижние колонтитулы документа. Если вы сейчас взглянете на верх этой страницы, то увидите одно из возможных применений этого пакета.

Сложность в настройке колонтитулов в том, чтобы включить туда вещи вроде заголовков раздела или главы. $\text{\LaTeX}$ достигает этого в два этапа. В определениях колонтитулов можно использовать команды `\rightmark` и `\leftmark`, представляющих заголовки текущей главы и раздела, соответственно. Значения этих двух команд меняются при обработке команд `\chapter` или `\section`.

Для большей гибкости команды `\chapter` и ей подобные не переопределяют `\rightmark` и `\leftmark` сами, а вызывают еще одну команду, называемую `\chaptermark`, `\sectionmark` или `\subsectionmark`, ответственную за переопределение `\rightmark` и `\leftmark`.

Так что, если вы хотите изменить вид названия главы в верхнем колонтитуле, вы просто переопределяете команду `\chaptermark`.

Рисунок 4.1 показывает, как можно настроить пакет `fancyhdr` так, чтобы колонтитулы выглядели почти так, как они выглядят у этого буклета. В любом случае вам рекомендуется ознакомиться с документацией к пакету по адресу, упомянутому в сноске.

¹Доступный из `/tex-archive/macros/latex/contrib/supported/fancyhdr`.

```

\documentclass{book}
\usepackage{fancyhdr}
\pagestyle{fancy}
% этим мы убеждаемся, что заголовки глав и
% разделов используют нижний регистр.
\renewcommand{\chaptermark}[1]{\markboth{#1}{}}
\renewcommand{\sectionmark}[1]{\markright{\thesection\ #1}}
\fancyhf{} % убираем текущие установки для колонтитулов
\fancyhead[LE,RO]{\bfseries\thepage}
\fancyhead[LO]{\bfseries\rightmark}
\fancyhead[RE]{\bfseries\leftmark}
\renewcommand{\headrulewidth}{0.5pt}
\renewcommand{\footrulewidth}{0pt}
\addtolength{\headheight}{0.5pt} % оставляем место для линейки
\fancypagestyle{plain}{%
  \fancyhead{} % на обычных страницах убираем колонтитулы
  \renewcommand{\headrulewidth}{0pt} % и линейку
}

```

Рис. 4.1: Пример настройки fancyhdr

4.5 Пакет verbatim

Ранее в этом введении вы познакомились с *окружением* `verbatim`. В этом разделе вы узнаете про *пакет* `verbatim`. Пакет `verbatim` представляет из себя повторную реализацию окружения `verbatim` с исправлением некоторых его ограничений. Само по себе это не замечательно,¹ но, кроме того, в него была добавлена некоторая функциональность, и вот почему пакет здесь упоминается. Пакет `verbatim` предоставляет команду

`\verbatiminput{файл}`

которая позволяет вам включать текстовый файл в ваш документ, как если бы его содержимое находилось внутри окружения `verbatim`.

Так как пакет `verbatim` входит в комплект ‘tools’, вы найдете его установленным на большинстве систем. Если вы хотите узнать о нем больше, обязательно прочитайте [10].

¹Это для американцев не замечательно. Для нас интересно то, что пакет `verbatim` включает команду `\verbatim@font`, позволяющую использовать произвольный шрифт, например, русский, что невозможно в окружении `verbatim` без модификации стандартного поведения ЛАТ_EX. — Прим. перев.

4.6 Загрузка и установка пакетов ЛАТ_EX

Большинство дистрибутивов ЛАТ_EX включают большой набор предустановленных стилевых пакетов, но в Сети доступно намного больше. Основное место для поиска стилевых пакетов в Интернете — CTAN (<http://www.ctan.org/>).

Пакеты, такие, как `geometry`, `hyphenat` и многие другие, обычно состоят из двух файлов: файла с расширением `.ins` и второго, с расширением `.dtx`. Часто к ним прилагается файл `readme.txt` с кратким описанием пакета. Конечно, сначала нужно прочесть этот файл.

В любом случае, после того, как вы скопировали файлы пакета на вашу машину, вам нужно обработать их так, чтобы, во-первых, ваш Т_EX узнал о наличии нового пакета и, во-вторых, вы получили необходимую документацию. Вот как выполняется первая часть:

1. Обработайте ЛАТ_EX файл `.ins`. На выходе вы получите файл `.sty`.
2. Поместите файл `.sty` туда, где ваш дистрибутив ищет эти файлы. Обычно это подкаталоги каталога `.../localtexmf/tex/latex`.
3. Обновите базу имен файлов вашего дистрибутива. Команда зависит от конкретного дистрибутива: для `teTeX` и `fpTeX` — `texhash`, для `web2c` — `maketexlsr`, для `MikTeX` — `initexmf -update-fndb`, или используя графические интерфейсы.

Теперь вы можете из файла `.dtx` получить документацию:

1. Обработайте ЛАТ_EX файл `.dtx`. На выходе вы получите файл `.dvi`. Заметьте, что вам может быть нужно запустить ЛАТ_EX несколько раз, чтобы получить правильные перекрестные ссылки.
2. Проверьте, не сгенерировал ли ЛАТ_EX, кроме всего прочего, файл `.idx`. Если нет, то переходите к пункту 5.
3. Для генерации предметного указателя, выполните команду:

`makeindex -s gind.ist имя`

(где *имя* — имя основного файла без расширения).
4. Снова обработайте ЛАТ_EX файл `.dtx`
5. В конце концов, для удобства чтения сгенерируйте файл `.ps` или `.pdf`.

Иногда вы обнаружите, что генерируется также файл `.glo` (гlossарий). В этом случае между шагами 4 и 5 выполните команду:

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

и еще раз запустить ЛАТ_EX с файлом `.dtx`.

4.7 Работа с pdf $\text{\LaTeX}$

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

PDF — формат гипертекстовых документов. Так же, как на веб-страницах, некоторые слова в документе отмечаются как гиперссылки. Они ведут к другим точкам документа или к другим документам. Если вы ткнете мышкой в такую гиперссылку, то попадете в точку назначения ссылки. В контексте $\text{\LaTeX}$ это означает, что все появления в тексте команд `\ref` и `\pageref` становятся гиперссылками. Кроме того, оглавление, предметный указатель и другие аналогичные структуры становятся наборами гиперссылок.

Большая часть веб-страниц сегодня пишется на языке HTML (*HyperText Markup Language*, *язык гипертекстовой разметки*). Для научных документов этот формат имеет два серьезных недостатка:

1. Включение в HTML математических формул, вообще говоря, не поддерживается. Несмотря на наличие такого стандарта, большинство используемых сегодня Интернет-навигаторов его не поддерживают или не имеют необходимых шрифтов.
2. Печать HTML-документов возможна, но результаты сильно зависят от используемой платформы и навигатора. Результат и близко не подходит к тому качеству, которое мы привыкли ожидать в мире $\text{\LaTeX}$.

Было множество попыток создать трансляторы из $\text{\LaTeX}$ в HTML. Некоторые из них были даже довольно успешны в том смысле, что они могут генерировать корректные веб-страницы из обычных файлов $\text{\LaTeX}$. Но все из них для получения результата предпринимают множество упрощений. Когда вы начинаете использовать сложные возможности $\text{\LaTeX}$ и внешние пакеты, такие трансляторы обычно начинают давать сбои. Авторы, желающие сохранить уникально типографское качество своих документов даже публикуясь на WWW, обращаются к PDF (*Portable Document Format*, *формат мобильных документов*), который сохраняет верстку документа и позволяет использовать гипертекстовую навигацию. Большинство современных веб-навигаторов имеют встроенные средства для просмотра документов в формате PDF.

Даже несмотря на наличие средств просмотра форматов DVI и PostScript почти на любой платформе, вы увидите, что гораздо более распространены Acrobat Reader и xpdf, служащие для просмотра PDF-документов. Поэтому, предоставляя PDF-версии ваших документов, вы делаете их более доступными для потенциальных читателей.

4.7.1 PDF-документы для WWW

Создание PDF-файла из исходного текста в $\text{\TeX}$ очень просто благодаря программе pdf $\text{\TeX}$, разработанной Hàn Th'ê Thành. pdf $\text{\TeX}$ генерирует PDF аналогично тому, как $\text{\TeX}$ генерирует DVI. Существует также pdf $\text{\LaTeX}$, генерирующий PDF из исходного теста $\text{\LaTeX}$.

И pdf $\TeX$, и pdf $\LaTeX$ автоматически устанавливаются большинством современных дистрибутивов $\TeX$, таких, как te $\TeX$, fp $\TeX$, Mik $\TeX$, $\TeX$ Live и CMac $\TeX$.

Для генерации PDF вместо DVI достаточно вместо команды `latex file.tex` использовать команду `pdflatex file.tex`. Там, где $\LaTeX$ не запускается из командной строки, вы можете найти специальную кнопку в $\TeX$ ControlCenter.

В $\LaTeX$ размер бумаги определяется при помощи необязательных параметров команды `\documentclass`, например, `a4paper` или `letterpaper`. Этот механизм работает и в pdf $\LaTeX$, но, помимо этого, pdf $\TeX$ должен знать также и физический размер бумаги, а не только размер области, используемой для верстки. Если вы используете пакет `hyperref` (см. стр. 77), размер бумаги будет установлен автоматически. В ином случае вам нужно сделать это вручную, поместив в преамбулу документа следующие строки:

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

Следующий раздел более детально поясняет разницу между «нормальным» $\LaTeX$ и pdf $\LaTeX$. Основные отличия сосредоточены в трех областях: в используемых шрифтах, в форматах включаемых изображений и в методах ручного формирования гиперссылок.

4.7.2 Шрифты

pdf $\LaTeX$ может работать с любыми видами шрифтов (растровые шрифты РК, TrueType, POSTSCRIPT type 1...), но основной для $\LaTeX$ формат, растровые шрифты РК, делают документ, просматриваемый на экране при помощи Acrobat Reader, очень коряво выглядящим. Для генерации нормально отображаемых документов лучше всего использовать шрифты POSTSCRIPT Type 1.

POSTSCRIPT Type 1 версии шрифтов Computer Modern и AMSFonts были произведены компаниями Blue Sky Research и Y&Y, Inc., которые потом передали авторские права на них Американскому Математическому Обществу (AMS). В начале 1997 года эти шрифты были сделаны публично доступными, и в настоящее время включены в большинство дистрибутивов $\TeX$.

Однако, если вы используете $\LaTeX$ для создания документов на языках, отличных от английского, вам может быть нужно использовать шрифты EC, LH или CB (см. обсуждение шрифтов OT1 на стр. ??). Владимир Волович создал шрифтовой пакет `cm-super`, который охватывает полные наборы шрифтов EC/TC, EC Concrete, EC Bright и LH. Он доступен по адресу fonts/ps-type1/cm-super, а также включен в дистрибутивы $\TeX$ Live7 и Mik $\TeX$. Аналогично, греческие шрифты CB в формате Type 1, созданные Apostolos Syropoulos, доступны по адресу fonts/greek/cb. К сожалению, оба эти набора не отличаются типографским качеством Type 1 CM шрифтов от Blue Sky/Y&Y. Хинты (hints) в них сгенерированы автоматически, и потому документы могут на экране выглядеть не так красиво, как при использовании Type 1 CM шрифтов Blue Sky/Y&Y. На устройствах с высоким разрешением они дают результаты, идентичные оригинальным растровым шрифтам EC/LH/CB.

Если вы создаете документы на латинице, у вас есть несколько других возможностей:

- Использовать пакет `aeguill`, известный также как *Almost European Computer Modern with Guillemets*. Просто поместите в преамбулу строчку `\usepackage{aeguill}` для использования виртуальных шрифтов АЕ вместо шрифтов ЕС.
- Использовать пакет `mltex`, который, однако, работает, только если ваш pdfTeX скомпилирован с опцией `mltex`.

Виртуальный набор шрифтов АЕ, аналогично системе MiTeX, заставляет TeX верить, что он имеет полные 256-символьные шрифты, создавая большинство недостающих знаков из шрифта CM и переупорядочивая их в кодировку ЕС, позволяя таким образом использовать высококачественные CM шрифты в формате Type 1, доступные на большинстве систем. Поскольку шрифт теперь имеет кодировку T1, для европейских языков, основанных на латинице, нормально работает механизм переносов. Единственный недостаток этого подхода — в том, что искусственные символы АЕ не работают в функции поиска Acrobat Reader, так что вы не сможете искать слова с акцентированными символами.

Для русского языка аналогичным решением будет использование виртуальных шрифтов C1, доступных по адресу <ftp://ftp.vsu.ru/pub/tex/font-packs/c1fonts>. Эти шрифты комбинируют стандартные шрифты CM Type 1 из набора Bluesky со шрифтами CMCYR Type 1 из наборов Paradissa и BaKoMa, доступных на CTAN. Поскольку шрифты Paradissa содержат только русские буквы, шрифты C1 не имеют других кириллических символов.

Иным решением может быть переключение на другие POSTSCRIPT Type 1 шрифты. Некоторые из них даже поставляются с каждой копией Acrobat Reader. Из-за того, что эти шрифты имеют другие метрики символов, ваша верстка изменится. Обычно текст начинает занимать больше места, так как CM шрифты очень компактны. Кроме того, общая визуальная согласованность документа будет хуже, так как шрифты Times, Helvetica и Courier (основные кандидаты на такую замену) не были спроектированы работать вместе в одном документе, как это было сделано в шрифтах Computer Modern.

Для этой цели существуют два готовых набора шрифтов: пакет `pxfonts`, базирующийся в качестве основного на шрифте *Palatino*, и пакет `txfonts`, базирующийся на шрифте *Times*. Чтобы использовать эти пакеты, поместите в преамбулу следующие строчки:

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

Замечание: после компиляции исходного текста вы можете увидеть в `.log`-файле строчки наподобие следующей:

Warning: pdftex (file eurmo10): Font eurmo10 at ... not found

Они означают, что не был найден определенный шрифт, использованный в документе. Вам придется устранить эту проблему, так как результирующий PDF-документ может вообще не показывать страницы с отсутствующими символами.

Вся эта непростая проблема со шрифтами, особенно отсутствие хорошего набора шрифтов ЕС, равного по качеству шрифтам CM в формате Type 1, занимает умы множества людей, так что постоянно появляются новые решения.

4.7.3 Использование графики

Включение графики в документ лучше всего работает при помощи пакета `graphicx` (см. стр. 67). С указанием значения `pdftex` в опции *driver* этот пакет успешно работает с pdfL^AT_EX:

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

В приведенном примере включен также пакет `color`, поскольку использование цвета в веб-документах вполне естественно.

Это все были хорошие новости, а теперь — плохие: графика в формате Encapsulated POSTSCRIPT не работает с pdfL^AT_EX. Если вы не зададите расширение файла в команде `\includegraphics`, пакет `graphicx` выберет подходящий файл, основываясь на значении опции *driver*. Для значения `pdftex` это будут форматы `.png`, `.pdf`, `.jpg`, `.mps` (METAPOST) и `.tif` — но не `.eps`.

Простой способ обхода этой проблемы — сконвертировать ваши файлы EPS в формат PDF утилитой `epstopdf`, присутствующей во многих системах. Для векторной графики это замечательное решение. Для растров (фотографий, сканированных документов) это не идеально, потому что формат PDF сам по себе поддерживает включение растровых изображений формата PNG и JPEG. Формат PNG хорош для снимков экрана и других изображений с небольшим количеством цветов. Формат JPEG хорош для фотографий из-за своей компактности.

Иногда может быть желательно вообще не рисовать геометрические фигуры, а описать их на языке специальных команд, таком, как METAPOST, включенном в большинство дистрибутивов T_EX. В комплекте вы найдете полное руководство по его использованию.

4.7.4 Гиперссылки

Пакет `hyperref` превратит все внутренние ссылки документа в гиперссылки. Для этого необходимо некоторое шаманство, в частности, команда `\usepackage[pdftex]{hyperref}` должна быть *последней* командой в преамбуле вашего документа.

Для настройки поведения пакета `hyperref` мы можете использовать:

- разделенный запятыми список опций `hyperref`, после опции `pdftex`
`\usepackage[pdftex]{hyperref}`

- отдельные команды `\hypersetup{опции}`.

Единственная обязательная опция — `pdftex`, остальные лишь позволяют изменять поведение `hyperref` по умолчанию.¹ В следующем списке значения по умолчанию приведены прямым шрифтом:

`bookmarks (=true, false)` показывает или прячет полосу закладок при просмотре документа

`unicode (=false, true)` позволяет использовать в закладках нелатинские символы

`pdftoolbar (=true, false)` показывает или прячет линейку инструментов Acrobat

`pdfmenubar (=true, false)` показывает или прячет меню Acrobat

`pdffitwindow (=true, false)` меняет начальное увеличение документа (страница размером в окно Acrobat)

`pdftitle ({text})` устанавливает заголовок, отображаемый Acrobat в окне Document Info

`pdfauthor ({text})` задает автора PDF-документа

`pdfnewwindow (=true, false)` определяет, должно ли открываться новое окно, если гиперссылка ведет в другой документ

`colorlinks (=false, true)` окружает ссылки цветными рамками (`false`) или меняет цвет текста ссылок (`true`). Цвет можно настроить следующими опциями (показаны значения по умолчанию):

`linkcolor (=red)` цвет внутренних ссылок (разделы, страницы и т.п.),

`citecolor (=green)` цвет ссылок на литературу,

`filecolor (=magenta)` цвет ссылок на файлы,

`urlcolor (=cyan)` цвет ссылок на URL (почта, WWW).

Если вас устраивают значения по умолчанию:

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

Чтобы открыть окно закладок и раскрасить ссылки (значения `=true` можно опускать):

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

¹Стоит заметить, что пакет `hyperref` не ограничен работой с `pdfTeX`. Его можно настроить для внедрения в нормальную выдачу DVI специфичной для PDF информации. Она потом попадает в PS-файл, генерируемый программой `dvips`, и, наконец, используется Adobe Distiller, который можно использовать для преобразования PS-файлов в формат PDF.

Когда PDF-документы предназначены для печати, цветные ссылки лучше не использовать, потому что они окажутся на бумаге серыми, затрудняя чтение. Лучше использовать цветные рамки, которые не печатаются:

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

или делать ссылки черными:

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

Если вам нужно задать информацию для секции Document Info:

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

В дополнение к автоматически генерируемым гиперссылкам, возможно вставлять ссылки в явной форме с помощью команды

`\href{url}{text}`

Код

```
Веб-сайт \href{http://www.ctan.org}{CTAN}.
```

генерирует на выходе “CTAN”; при нажатии мышью на слово “CTAN” вы попадете на веб-сайт CTAN.

Если ссылка ведет не на URL, а на локальный файл, вы тоже можете пользоваться командой `\href`:

```
Полный текст документа находится \href{manual.pdf}{здесь}.
```

что сгенерирует текст “Полный текст документа находится [здесь](#).” Нажатие на слово “здесь” откроет файл `manual.pdf`. (Имя файла рассматривается относительно положения текущего документа.)

Автор статьи может облегчить читателю отсылку откликов, используя команду `\href` внутри команды `\author` на титульной странице документа:

```
\author{Mary Oetiker $\<\href{mailto:mary@oetiker.ch}%
        {mary@oetiker.ch}$>$}
```

Заметим, что ссылка с почтовым адресом дополняет адрес, приведенный собственно на странице. Это сделано потому, что ссылка

```
\href{mailto:mary@oetiker.ch}{Mary Oetiker}
```

будет удобна из-под Acrobat Reader, но будет невидима после печати документа на бумаге.

4.7.5 Проблемы со ссылками

Когда сбрасываются счетчики, например, когда команда `\mainmatter` из класса документов `book` сбрасывает счетчик номера страницы в 1, генерируются сообщения наподобие такого:

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

Происходит это потому, что во вводной части книги уже была страница номер 1, и все ссылки на “страницу 1” будут уже не уникальными, отсюда и сообщение “duplicate has been ignored.”

Избавиться от этого можно, задав `hyperref` опцию `plainpages=false`. К сожалению, поможет она только в отношении счетчика страниц. Еще более радикальной мерой может быть использование опции `hypertexnames=false`, но после этого перестанут работать ссылки на страницы из предметного указателя.

4.7.6 Проблемы с закладками

Текст закладок не всегда выглядит так, как вам бы этого хотелось. В закладках допустим гораздо меньший набор символов, чем в нормальном $\text{\LaTeX}$, поскольку они являются “просто текстом”. Обычно `hyperref` замечает связанные с этим проблемы и выводит предупреждение:

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

Вы можете обойти эту проблему, предоставив текстовую строчку для закладки взамен проблемного текста:

```
\texorpdfstring{текст  $\text{\TeX}$ }{текст закладки}
```

Математические выражения являются первыми кандидатами на такую замену:

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

что превращает `\section{$E=mc^2$}` в “ $E=mc^2$ ” для цели создания закладки.

Цвета тоже плохо себя ведут в закладках:

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

выдает строку “redRed!”. Команда `\textcolor` игнорируется, но ее аргумент (red) печатается.

Гораздо лучший результат получается так:

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

4.8 Совместимость исходных текстов $\text{\LaTeX}$ и $\text{pdf\LaTeX}$

В идеале ваш документ должен одинаково хорошо компилироваться как $\text{\LaTeX}$, так и $\text{pdf\LaTeX}$. Основная проблема тут — включение графики. Простым решением является *повсеместное умолчание* расширения файлов в команде `\includegraphics`. В этом случае компилятор автоматически выбирает файл требуемого формата из текущего каталога. Все, что нужно будет сделать, — это создать подходящие версии графических файлов. $\text{\LaTeX}$ будет выбирать `.eps`, а $\text{pdf\LaTeX}$ постарается включить файлы с расширениями `.png`, `.pdf`, `.jpg`, `.mps` или `.tif` (в указанном порядке).

Для случаев, когда вам все же нужно будет использовать разный код для обычной и PDF-версии документа, можно где-то в самом начале документа сделать следующее:

```
\newif\ifPDF
\ifx\pdfoutput\undefined\PDFfalse
\else\ifnum\pdfoutput > 0\PDFtrue
  \else\PDFfalse
  \fi
\fi
```

Здесь определена специальная команда, позволяющая легко писать условный код:

```
\ifPDF
  \usepackage[T1]{fontenc}
  \usepackage{aeuill}
  \usepackage[pdftex]{graphicx,color}
  \usepackage[pdftex]{hyperref}
\else
  \usepackage[T1]{fontenc}
  \usepackage[dvips]{graphicx}
  \usepackage[dvips]{hyperref}
\fi
```

В этом примере пакет `hyperref` включен даже в не-PDF-версии. Сделано это для того, чтобы команда `\href` работала в любом случае, и ее использование не надо было оборачивать в условные операторы.

Заметим, что в современных дистрибутивах $\text{\TeX}$ (например, в $\text{\TeX Live}$), выбор `pdftex` или `dvips` при вызове `graphicx` и `color` происходит автоматически в соответствии с настройками конфигурационных файлов `graphics.cfg` и `color.cfg`.

4.9 Создание презентаций при помощи pdfscreen

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

Результаты вашей работы вы можете демонстрировать на доске, проектором, или прямо с вашего ноутбука при помощи одной из презентационных программ.

pdfL^AT_EX с пакетом **pdfscreen** позволяет вам создавать презентации в PDF, столь же красивые и живые, как и в *PowerPoint*, но много более мобильные, поскольку Acrobat Reader существует на более широком спектре систем.

Пакет **pdfscreen** использует пакеты **graphicx**, **color** и **hyperref** с опциями, настроенными для экранной презентации.

```
\documentclass[pdftex,12pt]{article}
%% misc extensions %%%%%%%%%%%%%%
\usepackage[latin1]{inputenc}
\usepackage[english]{babel}
\usepackage[T1]{fontenc}
\usepackage{aequill}
%% pdfscreen %%%%%%%%%%%%%%
\usepackage[screen,panelfleft,chocolate]{pdfscreen}
% Screen Format
\panelwidth=25mm
%%          height width
\screensize{150mm}{200mm}
%%          left right top bottom
\marginwidth{42mm}{8mm}{10mm}{10mm}
% Color or image for background
\overlayempty
\definecolor{mybg}{rgb}{1,0.9,0.7}
\backgroundcolor{mybg}
% Logo
\emblem{MyLogo}
%% For PPower4 (post-processor) %%%%%%%%%%
\usepackage{pause}
%%%%%%%%%
\begin{document}
\begin{slide}
\begin{itemize}
\item Good News\dots \pause
\item Bad News
\end{itemize}
\end{slide}
\end{document}
```

Рис. 4.2: Пример входного файла pdfscreen

Для создания таких документов обычно используется класс `article`. Рис. 4.2 показывает пример входного файла. Сначала загружается пакет `pdfscreen` с соответствующими опциями:

`screen` : экранная презентация. Для бумажной презентации пользуйтесь опцией `print`.

`panelright` размещает панель навигации в правой части экрана. Если панель нужна слева, пользуйтесь опцией `panelleft`. Если она не нужна вообще, используйте `nopanel`.

`french` или любой другой поддерживаемый язык. Текст на навигационных кнопках выводится на соответствующем языке. Эта опция не зависит от опций пакета `babel`. Если ваш язык не поддерживается `pdfscreen`, вы все же можете локализовать кнопки панели при помощи файла `pdfscreen.cfg`, см., например, `pdfscreen.cfg.specimen`.

`chocolate` цветовая схема навигационной панели. Другие варианты: `gray`, `orange`, `palegreen`, `bluelace` и `blue`, используемый по умолчанию.

Далее настраивается формат экрана. Поскольку презентация всегда разворачивается на полный экран, это можно использовать для настройки размера шрифта:

`\panelwidth` определяет ширину навигационной панели

`\screensize{ширина}{высота}` определяет ширину и высоту экрана, включая навигационную панель

`\marginsize{левое}{правое}{верхнее}{нижнее}` определяет поля документа. В приведенном примере документ не центрируется, поскольку номера разделов остаются на левом поле.

Можно использовать фоновое изображение в любом из поддерживаемых pdfTeX форматов командой

```
\overlay{изображение}
```

или, если вы предпочитаете пустой фон, можно задать его цвет командой

```
\background{цвет}
```

Наконец, если вы хотите поместить на навигационную панель эмблему вашей организации, пользуйтесь командой

```
\emblem{файл-с-эмблемой}
```

Если вы верите в убедительность последовательной демонстрации пунктов вашего рассказа, можете использовать пакет `pause`. Он предлагает команду `\pause`. Ее

можно помещать в текст в те места, где вы хотите задержать отображение вашего документа. Пакет `pause` является частью системы **PPower4** (*P⁴: PDF Presentation Post-Processor*), которая берет документ, сгенерированный pdfTeX, и заставляет его петь, плясать и собирать деньги. Командная строка выглядит так:

```
ppower4 xy.pdf xyz.pdf
```

Для управления тем, что отражается на каждом слайде, пользуйтесь окружением `\begin{slide} ... \end{slide}`. Содержимое каждого слайда отображается вертикально центрированным на своей странице.

Компилируя приведенный пример, мы получим ошибку:

```
! pdfTeX warning (dest): name{contents} has been
referenced but does not exist, replaced by a fixed one
```

Причина тут в том, что на навигационной панели есть кнопка, ведущая к оглавлению, а наш пример оглавления не имеет.

Если вам захочется отображать оглавление прямо на навигационной панели, используйте `pdfscreen` с опцией `paneltoc`. Понятно, это будет удовлетворительно выглядеть только если ваше оглавление содержит немного коротких заголовков. Можно задавать дополнительные короткие заголовки для ваших разделов квадратными скобками в командах секционирования.

Это короткое введение пробежало только по самым верхам возможностей `pdfscreen` и **PPower4**. Каждый из них включает собственное подробное руководство.

5 Генерация математической графики

Большинство людей пользуются $\text{\LaTeX}$ для верстки текста. Но $\text{\LaTeX}$ также предлагает, хотя и ограниченную, возможность генерации графики по текстовому описанию. Более того, существует множество расширений $\text{\LaTeX}$, обходящих эти ограничения. Эта глава расскажет о нескольких из них.

5.1 Обзор

Окружение `picture` позволяет программировать картинки прямо в среде $\text{\LaTeX}$. Подробное описание приводится в *$\text{\LaTeX}$ Manual* [1]. С одной стороны, при этом есть довольно серьезные ограничения, поскольку как наклоны отрезков, так и радиусы дуг сильно ограничены в возможных значениях. С другой стороны, в окружении `picture` существует команда `\qbezier`, где «q» означает «квадратичный». Множество часто используемых кривых, таких как окружности, эллипсы или цепные линии, можно удовлетворительно аппроксимировать квадратичными кривыми Безье, хотя для этого и требуются некоторые математические усилия. Кроме того, когда блоки команд `\qbezier` генерируются из языка программирования, скажем, Java, окружение `picture` становится довольно мощным.

Хотя программирование картинок прямо из $\text{\LaTeX}$ серьезно ограничено и часто довольно утомительно, оно все же бывает полезным. Полученные таким образом документы получаются компактными, и нет необходимости прилагать к документу графические файлы.

Пакеты наподобие `epic` и `eeepic` (описанные, например, в *The $\text{\LaTeX}$ Companion* [3]) или `pstricks` помогают снять ограничения оригинального окружения `picture` и сильно развивают графические возможности $\text{\LaTeX}$.

Тогда как первые два пакета просто улучшают окружение `picture`, пакет `pstricks` имеет собственное окружение для рисования, `pspicture`. Возможности `pstricks` проистекают из того, что он широко использует функциональность `POSTSCRIPT`. Кроме этого пакета, существует и множество других, написанных для конкретных задач. Одним из них является пакет `Xy-pic`, описанный в конце этой главы. Множество таких пакетов детально описано в *The $\text{\LaTeX}$ Graphics Companion* [4] (не путать с *The $\text{\LaTeX}$ Companion* [3]).

Возможно, самым мощным графическим инструментом, связанным с $\text{\LaTeX}$, является `METAPOST`, программа-близнец написанной Donald E. Knuth программы `METAFONT`. `METAPOST` использует очень мощный и математически изощренный язык

METAFONT. В отличие же от METAFONT, генерирующего растры, METAPOST генерирует файлы Encapsulated POSTSCRIPT, которые можно импортировать в L^AT_EX. Далее смотрите *Руководство пользователя MetaPost* [15], или введение в [17].

Очень подробно работа с графикой (и шрифтами) в L^AT_EX и T_EX описана в *T_EX Unbound* [16].

5.2 Окружение picture

Автор: Urs Oswald <osurs@bluewin.ch>

5.2.1 Основные команды

Окружение `picture` создается одной из двух команд:

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

либо

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

Числа x , y , x_0 , y_0 измеряются в размерности `\unitlength`, которую можно менять в любой момент (но не внутри окружения `picture`) командами наподобие

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

Значение `\unitlength` по умолчанию составляет 1pt. Первая пара, (x, y) , диктует резервирование для картинки прямоугольного пространства внутри документа. Необязательная вторая пара, (x_0, y_0) , присваивает произвольные координаты нижнему левому углу зарезервированного прямоугольника.

Большинство команд рисования имеют одну из двух форм:

```
\put(x,y){объект}
```

или

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

Кривые Безье являются исключением. Их рисуют командой

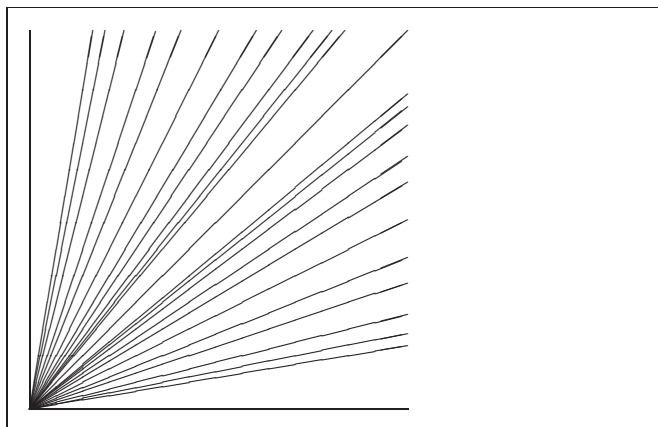
```
\qbezier(x_1,y_1)(x_2,y_2)(x_3,y_3)
```

5.2.2 Отрезки

```

\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}

```



Отрезки рисуются командой

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

Команда `\line` имеет два аргумента:

1. вектор направления,
2. длина.

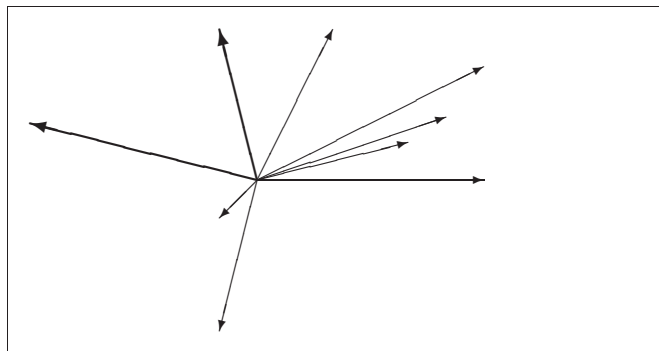
Компоненты вектора ограничены в своем значении набором целых чисел

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

и они обязаны быть взаимно простыми (не иметь общего делителя, кроме 1). Иллюстрация показывает все 25 возможных значений наклона в первом квадранте. Длина выражается в единицах `\unitlength`. Аргумент длины задает вертикальную координату в случае вертикального отрезка и горизонтальную — во всех остальных случаях.

5.2.3 Векторы

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



Векторы рисуются командой

$\text{\put}(x,y)\{\text{\vector}(x_1,y_1)\{length\}\}$

Для векторов значения вектора направления еще более ограничены в значениях, чем для отрезков, а именно — числами

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

Компоненты также обязаны быть взаимно простыми (не иметь общего делителя, кроме 1). Заметьте эффект команды `\thicklines` на двух векторах, указывающих в верхний левый угол.

5.2.4 Окружности

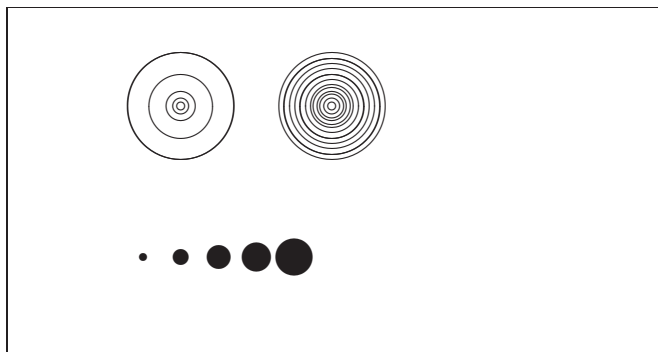
```

\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}

```



Команда

<code>\put(x,y){\circle{диаметр}}</code>
--

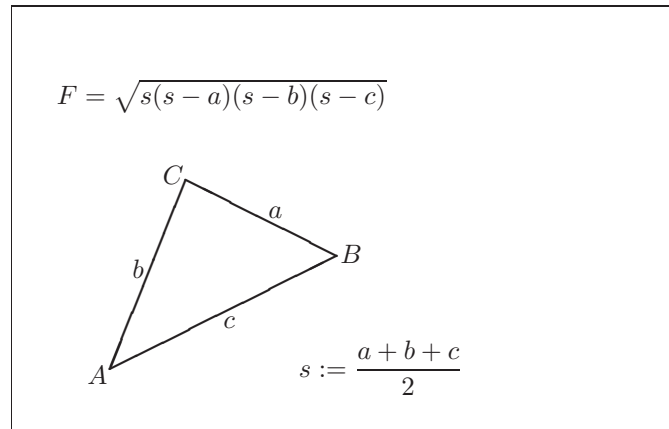
рисует окружность с центром в точке (x, y) и диаметром (не радиусом!) *диаметр*. Окружение `picture` позволяет рисовать окружности диаметром не более примерно 14 мм, и даже в этих пределах допустимы не все диаметры. Команда `\circle*` рисует круг (заполненную окружность).

Как и в случае отрезков, вы можете прибегать к помощи дополнительных пакетов, таких, как `eeepic` или `pstricks`. Подробное описание этих пакетов приведено в *The L^AT_EX Graphics Companion* [4].

Существует также и выход в рамках окружения `picture`. Если вы не боитесь выполнения необходимых вычислений (или возлагаете их на программу), то можно изображать произвольные окружности и эллипсы при помощи кривых Безье. Примеры и исходные тексты на Java приведены в *Графика в L^AT_EX 2_ε* [17].

5.2.5 Текст и формулы

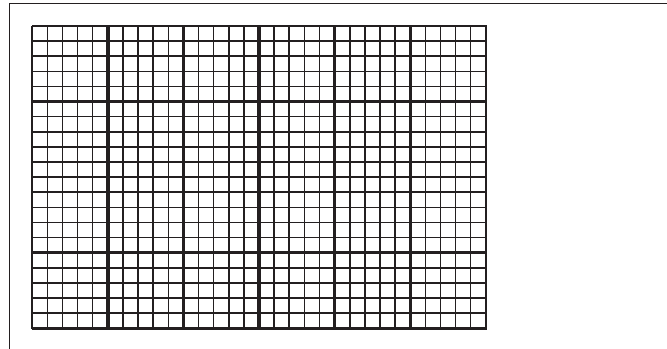
```
\setlength{\unitlength}{1cm}
\begin{picture}(6,5)
  \thicklines
  \put(1,0.5){\line(2,1){3}}
  \put(4,2){\line(-2,1){2}}
  \put(2,3){\line(-2,-5){1}}
  \put(0.7,0.3){$A$}
  \put(4.05,1.9){$B$}
  \put(1.7,2.95){$C$}
  \put(3.1,2.5){$a$}
  \put(1.3,1.7){$b$}
  \put(2.5,1.05){$c$}
  \put(0.3,4){$F=$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



Как показывает этот пример, текст и формулы могут размещаться в окружении `picture` обычным способом — командой `\put`.

5.2.6 Команды `\multiput` и `\linethickness`

```
\setlength{\unitlength}{2mm}
\begin{picture}(30,20)
  \linethickness{0.075mm}
  \multiput(0,0)(1,0){31}%
    {\line(0,1){20}}
  \multiput(0,0)(0,1){21}%
    {\line(1,0){30}}
  \linethickness{0.15mm}
  \multiput(0,0)(5,0){7}%
    {\line(0,1){20}}
  \multiput(0,0)(0,5){5}%
    {\line(1,0){30}}
  \linethickness{0.3mm}
  \multiput(5,0)(10,0){3}%
    {\line(0,1){20}}
  \multiput(0,5)(0,10){2}%
    {\line(1,0){30}}
\end{picture}
```



Команда

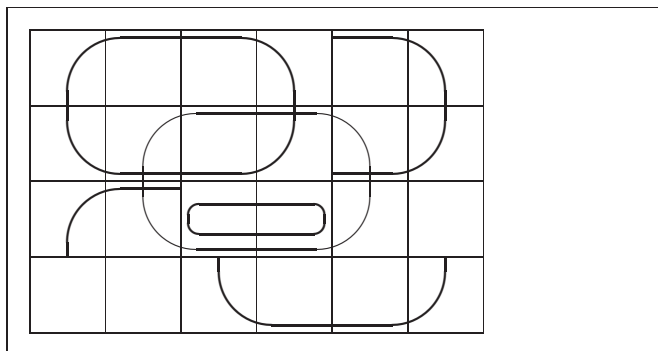
$\text{\multiput}(x, y)(\Delta x, \Delta y)\{n\}\{\text{объект}\}$

имеет 4 аргумента: начальная точка, вектор перехода от одного объекта к следую-

щему, число объектов и собственно объект для рисования. Команда `\linethickness` относится к горизонтальным и вертикальным отрезкам, но никогда — к наклонным или окружностям. Она, однако относится также и к квадратичным кривым Безье!

5.2.7 Овалы. Команды `\thinlines` и `\thicklines`

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



Команда

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

или

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

выводит овал с центром в (x, y) , имеющий длину w и высоту h . Необязательный аргумент *позиция* может принимать значения **b**, **t**, **l** и **r** (вниз/вверх/влево/вправо) и может сочетать эти значения, как на приведенном примере.

Толщина линий может управляться одной из двух команд:

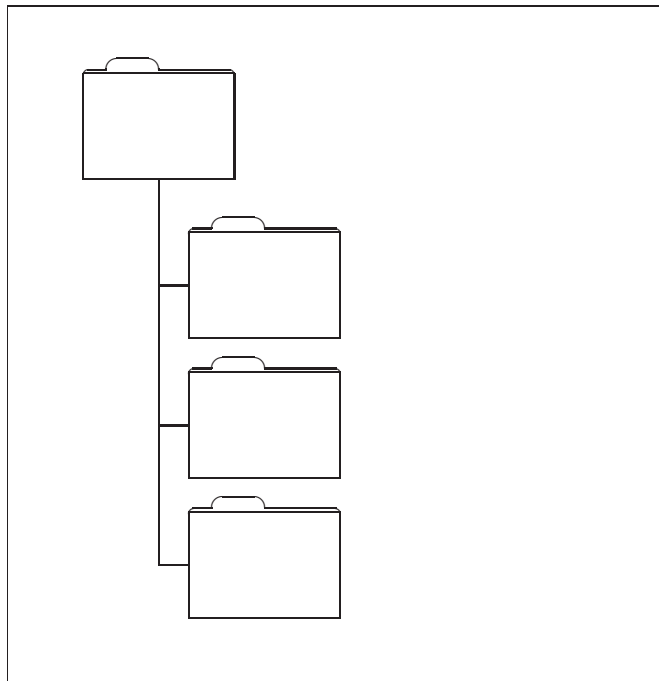
`\linethickness{длина}` с одной стороны, и `\thinlines` и `\thicklines` — с другой. В то время, как `\linethickness{длина}` влияет только на горизонтальные и вертикальные линии (и квадратичные кривые Безье), `\thinlines` и `\thicklines` влияет на наклонные отрезки, окружности и овалы.

5.2.8 Повторное использование блоков картинки

```

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

```



Блок рисунка может быть *объявлен* командой

```
\newsavebox{название}
```

а затем *определен* командой

```
\savebox{название}(ширина,высота)[позиция]{содержание}
```

и, наконец, сколько угодно раз *нарисован* командой

```
\put(x,y)\usebox{название}
```

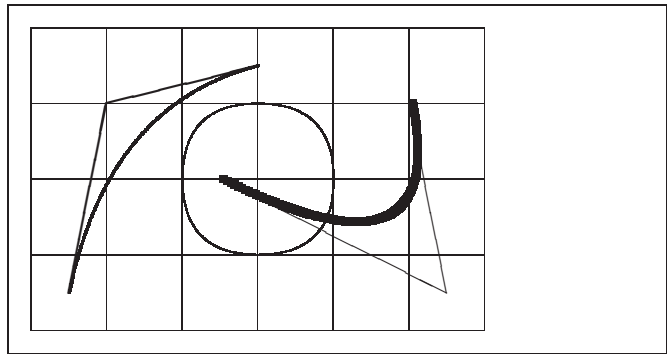
Необязательный аргумент *позиция* определяет точку привязки блока. В приведенном примере он установлен в значение **bl**, что помещает точку привязки в нижний левый угол блока. Другие варианты значения — **t** (вверх) и **r** (вправо).

Аргумент *название* фактически является командой $\LaTeX$ (отсюда и обратная косая черта перед ним в рассматриваемом примере). Блоки могут быть вложенными: в этом примере внутри определения `\folderb` используется `\foldera`.

Команду `\oval` пришлось применить потому что команда `\line` не работает, если длина отрезка меньше примерно 3 мм.

5.2.9 Квадратичные кривые Безье

```
\setlength{\unitlength}{1cm}
\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}
```



Как показывает этот пример, разбиение окружности на четыре квадратичных кривых Безье дает неудовлетворительный результат. Требуется как минимум восемь. Иллюстрация снова показывает влияние команды `\linethickness` на горизонтальные и вертикальные линии, а команд `\thinlines` и `\thicklines` — на наклонные отрезки. Она также показывает, что обе команды влияют на квадратичные кривые Безье, и каждая следующая команда отменяет результаты предыдущих.

Пусть $P_1 = (x_1, y_1)$, $P_2 = (x_2, y_2)$ задают конечные точки, а m_1, m_2 — соответствующие наклоны квадратичной кривой Безье. Тогда промежуточная управляющая точка $S = (x, y)$ задается уравнением

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

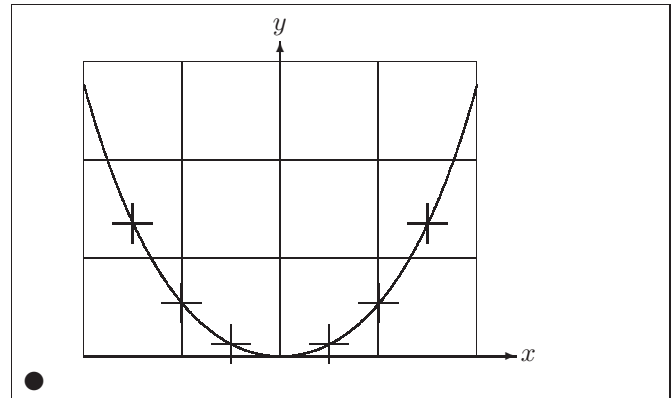
В *Графика в $\LaTeX$ 2_ε* [17] приведена Java-программа, генерирующая необходимые команды `\qbezier`.

5.2.10 Цепная линия

```

\setlength{\unitlength}{1.3cm}
\begin{picture}(4.3,3.6)(-2.5,-0.25)
  \put(-2,0){\vector(1,0){4.4}}
  \put(2.45,-.05){$x$}
  \put(0,0){\vector(0,1){3.2}}
  \put(0,3.35){\makebox(0,0){$y$}}
  \qbezier(0.0,0.0)(1.2384,0.0)
    (2.0,2.7622)
  \qbezier(0.0,0.0)(-1.2384,0.0)
    (-2.0,2.7622)
  \linethickness{.075mm}
  \multiput(-2,0)(1,0){5}
    {\line(0,1){3}}
  \multiput(-2,0)(0,1){4}
    {\line(1,0){4}}
  \linethickness{.2mm}
  \put(.3,.12763){\line(1,0){.4}}
  \put(.5,-.07237){\line(0,1){.4}}
  \put(-.7,.12763){\line(1,0){.4}}
  \put(-.5,-.07237){\line(0,1){.4}}
  \put(.8,.54308){\line(1,0){.4}}
  \put(1,.34308){\line(0,1){.4}}
  \put(-1.2,.54308){\line(1,0){.4}}
  \put(-1,.34308){\line(0,1){.4}}
  \put(1.3,1.35241){\line(1,0){.4}}
  \put(1.5,1.15241){\line(0,1){.4}}
  \put(-1.7,1.35241){\line(1,0){.4}}
  \put(-1.5,1.15241){\line(0,1){.4}}
  \put(-2.5,-0.25){\circle*{0.2}}
\end{picture}

```



На этой иллюстрации каждая симметричная половина цепной линии $y = \cosh x - 1$ аппроксимирована квадратичной кривой Безье. Правая половина кривой заканчивается в точке $(2, 2.7622)$, наклон в которой имеет значение $m = 3.6269$. Вновь используя уравнение (5.1), мы можем вычислить внутренние управляющие точки. Ими оказались $(1.2384, 0)$ и $(-1.2384, 0)$. Крестики отмечают точки *настоящей* цепной линии. Ошибка едва заметна, будучи меньше одного процента.

Этот пример показывает использование необязательного аргумента команды `\begin{picture}`. Картинка определена в удобных «математических» координатах, тогда как команда

```
\begin{picture}(4.3,3.6)(-2.5,-0.25)
```

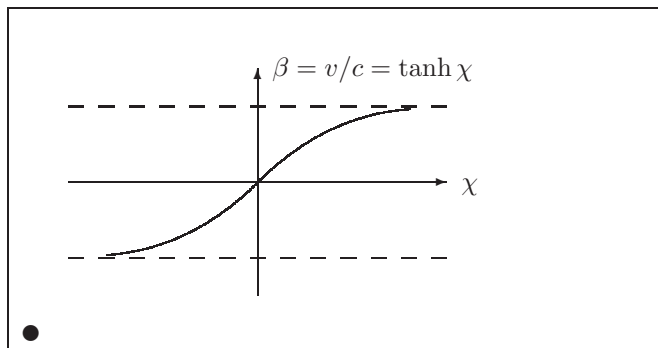
присваивает ее нижнему левому углу (отмеченному черным кружком) координаты $(-2.5, -0.25)$.

5.2.11 Скорость в специальной теории относительности

```

\setlength{\unitlength}{1cm}
\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}

```



Управляющие точки двух кривых Безье были вычислены по формулам (5.1). Положительная ветка определяется $P_1 = (0, 0)$, $m_1 = 1$ и $P_2 = (2, \tanh 2)$, $m_2 = 1/\cosh^2 2$. Картинка вновь определяется в математически удобных координатах, а нижний левый угол получает математические координаты $(-3, -2)$ (черный кружок).

5.3 Xy-pic

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

xy — это специальный пакет для рисования диаграмм. Для его использования просто добавьте к преамбуле документа следующую строчку:

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

где *onxy* — список загружаемых функций Xy-pic. Эти опции полезны, в первую очередь, для отладки пакета. Рекомендуется использовать опцию **all**, инструктируя L^AT_EX загрузить все команды Xy.

Диаграммы Xy-pic рисуются в матричной канве, где каждый элемент диаграммы помещается в определенную клетку матрицы:

```

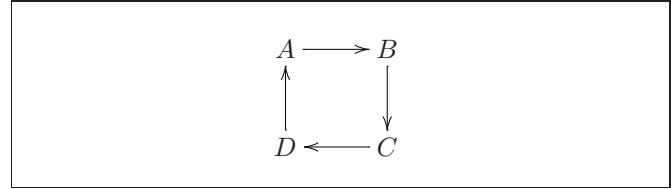
\begin{displaymath}
\xymatrix{A & B \\
          C & D }
\end{displaymath}

```

A	B
C	D

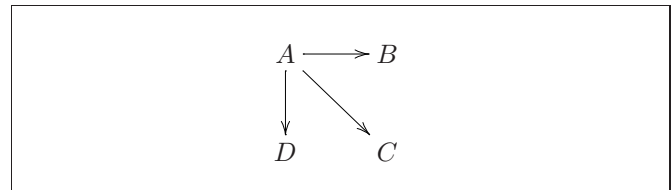
Команда `\xymatrix` должна использоваться в математическом режиме. Здесь мы задали две строки и два столбца. Чтобы из этой матрицы сделать диаграмму, добавим стрелки векторов командой `\ar`.

```
\begin{displaymath}
\xymatrix{ A \ar[r] & B \ar[d] \\
          D \ar[u] & C \ar[l] }
\end{displaymath}
```



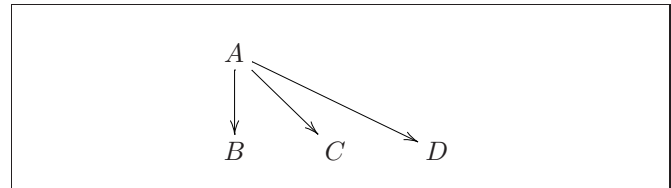
Команда рисования вектора помещается в клетку, откуда исходит вектор. Аргументом является направление, куда показывает вектор (up, down, right и left).

```
\begin{displaymath}
\xymatrix{
A \ar[d] \ar[dr] \ar[r] & B \\
D & C }
\end{displaymath}
```



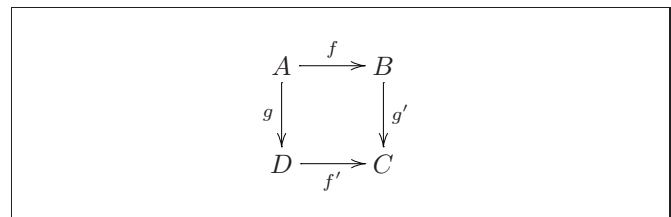
Для рисования диагоналей укажите больше одного направления. Вы также можете повторять знак направления для рисования больших векторов.

```
\begin{displaymath}
\xymatrix{
A \ar[d] \ar[dr] \ar[dr] & & \\
B & & C & D }
\end{displaymath}
```



Можно рисовать еще более интересные диаграммы, добавляя к векторам метки. Для этого используются обычные операторы нижних и верхних индексов.

```
\begin{displaymath}
\xymatrix{
A \ar[r]^f \ar[d]_g & B \\
D \ar[r]_{f'} \ar[d]_{g'} & C }
\end{displaymath}
```



Как показано выше, эти операторы используются как в математическом режиме. Единственная разница заключается в том, что верхний индекс означает «над стрелкой вектора», а нижний — «под стрелкой». Есть еще третий оператор, вертикальная черта: `|`. Он помещает текст *в* стрелку.

```

\begin{displaymath}
\xymatrix{
  A \ar[r]|f \ar[d]|g & \\
    B \ar[d]|{g'} \\\
  D \ar[r]|{f'} & C }
\end{displaymath}

```

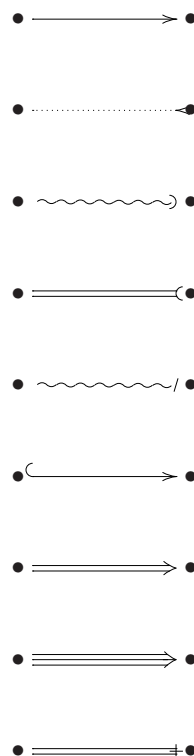
$$\begin{array}{ccc}
 A & \xrightarrow{f} & B \\
 \downarrow g & & \downarrow g' \\
 D & \xrightarrow{f'} & C
 \end{array}$$

Чтобы нарисовать стрелку с пробелом в ней, пользуйтесь командой `\ar[...]\hole`. В некоторых случаях важно различать несколько видов стрелок. Этого можно добиться, помещая на них метки или меняя их вид:

```

\shorthandoff{"}
\begin{displaymath}
\xymatrix{
  \bullet \ar@{->}[rr] && \bullet \\
  \bullet \ar@{.<}[rr] && \bullet \\
  \bullet \ar@{~}[rr] && \bullet \\
  \bullet \ar@{=(}[rr] && \bullet \\
  \bullet \ar@{~/}[rr] && \bullet \\
  \bullet \ar@{~{(->}}[rr] && \bullet \\
  \bullet \ar@2{->}[rr] && \bullet \\
  \bullet \ar@3{->}[rr] && \bullet \\
  \bullet \ar@{=+}[rr] && \bullet
}
\end{displaymath}
\shorthandon{"}

```

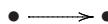


Отметьте разницу между следующими двумя диаграммами:

```

\begin{displaymath}
\xymatrix{
  \bullet \ar[r]
    \ar@{.>}[r] &
  \bullet
}
\end{displaymath}

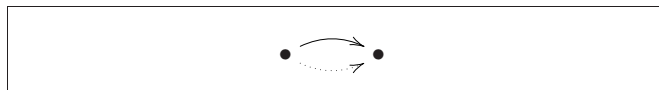
```



```

\begin{displaymath}
\begin{xy}
\bullet \ar@{~}/[r]
\bullet \ar@/_/@{.>}[r] &
\bullet
\end{xy}
\end{displaymath}

```



Модификаторы между косыми чертами определяют, как будут рисоваться кривые. $\text{\texttt{Xy-pic}}$ предлагает множество способов изменить способ рисования кривых; подробности смотрите в документации на $\text{\texttt{Xy-pic}}$.

6 Настройка $\text{\LaTeX}$

Произведенные с использованием изученных до сих пор команд документы будут выглядеть вполне приемлемо для широкой аудитории. Они не выглядят очень модно, подчиняясь зато всем правилам хорошего тона в наборе, поэтому их легко читать и на них приятно смотреть.

Но бывают ситуации, в которых $\text{\LaTeX}$ не предоставляет команду или окружение, удовлетворяющие вашим ожиданиям, или производимый некоторой существующей командой вывод не отвечает вашим требованиям.

В этой главе даются некоторые советы по обучению $\text{\LaTeX}$ новым вещам, и тому, как сделать его вывод отличающимся от того, что производится по умолчанию.

6.1 Новые команды, окружения и пакеты

Как вы заметили, все вводимые в этой книге команды верстаются в рамке и включаются в указатель в конце книги. Вместо того, чтобы напрямую использовать необходимые для этого команды $\text{\LaTeX}$, автор создал пакет, в котором определил новые команды и окружения для этой цели. Теперь можно просто писать:

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



В этом примере используются как новое окружение, называющееся `lscommand` и отвечающее за рисование рамки вокруг команды, так и новая команда, называющаяся `\ci` и верстающая название команды и заносящая соответствующий элемент в указатель. Вы можете в этом убедиться, поискав команду `\dum` в указателе в конце книги, где вы найдете запись для `\dum`, указывающую на эту страницу.

Если автор когда-нибудь решит, что ему не нравятся больше команды, сверстаные в рамке, он просто изменит определение окружения `lscommand`. Это намного проще, чем пройти по всему документу, выискивая все места, где использованы общие команды $\text{\LaTeX}$ для рисования рамки вокруг слов.

6.1.1 Новые команды

Чтобы добавить ваши собственные команды, пользуйтесь командой

`\newcommand{название}[число]{определение}`

Обычно эта команда требует двух аргументов. *Название* команды, которую вы создаете, и *определение* команды. Аргумент *число* в квадратных скобках не обязателен. Он применяется для создания новых команды, которые, в свою очередь, принимают до 9 аргументов.

Следующие два примера должны вам помочь получить представление о команде. Первый пример определяет новую команду, называющуюся `\tnss`, что является сокращением от “The Not So Short Introduction to ЛАТЭХ 2_ε”. Такая команда пригодится, если вам много раз приходится писать название этой книги.

```
\newcommand{\tnss}{The not
  so Short Introduction to
  \LaTeXe}
% в теле документа:
‘‘\tnss’’ \ldots{ } ‘‘\tnss’’
```

“The not so Short Introduction to ЛАТЭХ 2_ε” ... “The not so Short Introduction to ЛАТЭХ 2_ε”

Следующий пример показывает, как определить новую команду, принимающую один аргумент. Метка **#1** заменяется на заданный аргумент. Если вы хотите использовать более одного аргумента, пользуйтесь **#2**, и так далее.

```
\newcommand{\txsit}[1]
  {\emph{#1} краткое
   введение в \LaTeXe}
% в теле документа
\begin{itemize}
\item \txsit{Не очень}
\item \txsit{Очень}
\end{itemize}
```

- *Не очень* краткое введение в ЛАТЭХ 2_ε
 - *Очень* краткое введение в ЛАТЭХ 2_ε

ЛАТЭХ не позволит вам создать новую команду, которая бы изменяла уже существующую. Но для случая, когда вы явно хотите изменить существующую команду, есть специальная команда: `\renewcommand`. Она имеет тот же синтаксис, что и команда `\newcommand`.

В некоторых случаях может пригодиться команда `\providecommand`. Она работает так же, как `\newcommand`, но, если команда уже определена, то ЛАТЭХ 2_ε ее молча проигнорирует.

Существуют определенные особенности, связанные с пробелами после команд ЛАТЭХ. Подробности смотрите на странице 5.

6.1.2 Новые окружения

Аналогичная команде `\newcommand`, существует команда для создания вашего собственного окружения, `\newenvironment`, имеющая следующий синтаксис:

```
\newenvironment{название}[номер]{начало}{конец}
```

Подобно команде `\newcommand`, `\newenvironment` можно использовать с необязательным аргументом, или без него. Материал, заключенный в аргумент *начало*, обрабатывается до обработки текста внутри окружения. Материал, заключенный в аргумент *конец*, обрабатывается, когда встречается команда `\end{название}`. Следующий пример иллюстрирует использование команды `\newenvironment`.

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

\begin{king}
Мои смиренные подданные\ldots
\end{king}
```

■ Мои смиренные подданные... ■

Аргумент *номер* используют так же, как и для команды `\newcommand`. $\text{\LaTeX}$ контролирует, чтобы вы не определяли уже существующее окружение. Если вы заходите все же это сделать, пользуйтесь командой `\renewenvironment`. Она имеет тот же синтаксис, что и `\newenvironment`.

Команды, использованные в этом примере, будут разъяснены позже: описание команды `\rule` см. на стр. 113, команда `\stretch` описана на стр. 107, а описание команды `\hspace` находится на стр. 107.

6.1.3 Ваш собственный пакет

Когда вы определяете множество новых окружений и команд, преамбулы ваших документов становятся очень большими. В этой ситуации представляется разумным создать пакет $\text{\LaTeX}$, содержащий определения всех ваших команд и окружений. Потом можно командой `\usepackage` использовать пакет в ваших документах.

Создание пакета в основном состоит из переноса содержимого вашей преамбулы в отдельный файл с именем, заканчивающимся на `.sty`. Есть только одна специальная команда, которую вы должны использовать

```
\ProvidesPackage{название пакета}
```

в самом начале файла с вашим пакетом. `\ProvidesPackage` указывает $\text{\LaTeX}$ название пакета, что позволяет ему выдавать осмысленное сообщение об ошибке, когда вы

```
% Пакет для демонстрации. Tobias Oetiker.
\ProvidesPackage{demopack}
\newcommand{\tnss}{Не очень краткое введение в \LaTeXe}
\newcommand{\txsit}[1]{\emph{#1} краткое
                      введение в \LaTeXe}
\newenvironment{king}{\begin{quote}}{\end{quote}}
```

Рис. 6.1: Пример пакета

пытаетесь включать пакет дважды. Иллюстрация 6.1 показывает маленький пример пакета, содержащего определенные в вышеприведенных примерах команды.

6.2 Шрифты и их размеры

6.2.1 Команды смены шрифта

ЛАТЭХ выбирает подходящее начертание и размер шрифта, основываясь на логической структуре документа (разделы, сноски, ...). Иногда может быть желательно сменить шрифт вручную. Для этого вы можете пользоваться командами, перечисленными в таблицах 6.1 и 6.2. Действительный размер каждого шрифта определяется дизайном и зависит от класса и опций документа. Таблица 6.3 показывает абсолютные размеры, соответствующие этим командам в стандартных классах документов.

```
{\small Маленький,
 \textbf{полужирный},
 \Large большой,
 \textit{курсив}.}
```

Маленький, полужирный , БОЛЬШОЙ, <i>курсив</i> .

Важная особенность ЛАТЭХ 2_ε заключается в том, что атрибуты шрифта независимы. Это значит, что вы можете давать команды смены размера или даже семейства шрифта, сохраняя при этом установки атрибутов наклона или насыщенности.

В *математическом режиме* вы можете использовать *команды* смены шрифта, чтобы временно выйти из *математического режима* и ввести нормальный текст. Если вы хотите переключиться на другой шрифт для верстки математики, то для этого существует отдельный набор команд. Смотрите таблицу 6.4.

В связи с командами смены размера шрифта заметную роль играют фигурные скобки. Они используются для построения *групп*. Группы ограничивают область действия большинства команд ЛАТЭХ.

```
Ему нравятся {\LARGE
               большие и {\small
               маленькие} буквы}.
```

Ему нравятся БОЛЬШИЕ И маленькие БУКВЫ.

Таблица 6.1: Шрифты

<code>\textrm{...}</code>	прямой шрифт	<code>\textsf{...}</code>	без засечек
<code>\texttt{...}</code>	пишущая машинка		
<code>\textmd{...}</code>	нормальный	<code>\textbf{...}</code>	полужирный
<code>\textup{...}</code>	прямой шрифт	<code>\textit{...}</code>	<i>курсив</i>
<code>\textsl{...}</code>	<i>наклонный шрифт</i>	<code>\textsc{...}</code>	КАПИТЕЛЬ
<code>\emph{...}</code>	<i>выделенный шрифт</i>	<code>\textnormal{...}</code>	обычный

Таблица 6.2: Размеры шрифта

<code>\tiny</code>	крошечный	<code>\Large</code>	еще больше
<code>\scriptsize</code>	очень маленький	<code>\LARGE</code>	очень большой
<code>\footnotesize</code>	довольно маленький	<code>\huge</code>	огромный
<code>\small</code>	маленький	<code>\Huge</code>	громадный
<code>\normalsize</code>	нормальный		
<code>\large</code>	большой		

Таблица 6.3: Абсолютные размеры шрифтов в стандартных классах

Размер	10pt (по умолчанию)	опция 11pt	опция 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

Команды, влияющие на размер шрифта, влияют также на расстояние между строками, но только если соответствующий абзац заканчивается внутри области действия команды. Поэтому закрывающая фигурная скобка `}` не должна стоять слишком рано. Заметьте положение команды `\par` в следующих двух примерах¹:

```
{\Large Не читайте это! Это
неправда. Верьте мне!\par}
```

Не читайте это! Это неправда. Верьте мне!

```
{\Large Это тоже неправда.
Но помните, что я вру.\par}
```

Это тоже неправда. Но помните, что я вру.

Если вы хотите применить команду изменения размера к целому абзацу текста или больше того, то для этого лучше использовать синтаксис окружения.

```
\begin{Large}
  Это неправда. Но
  что в наши дни\ldots
\end{Large}
```

Это неправда. Но что в наши дни...

Это избавит вас от подсчета множества фигурных скобок.

6.2.2 Опасность!

Как отмечено в начале этой главы, опасно усеивать ваши документы явными командами, вроде только что описанных, потому что это противоречит основной идее

¹`\par` эквивалентен пустой строке.

Таблица 6.4: Математические шрифты

Команда	Пример	Вывод
<code>\mathcal{...}</code>	<code>\$\mathcal{B}=c\$</code>	$\mathcal{B} = c$
<code>\mathrm{...}</code>	<code>\$\mathrm{K}_2\$</code>	K_2
<code>\mathbf{...}</code>	<code>\$\sum x=\mathbf{v}\$</code>	$\sum x = \mathbf{v}$
<code>\mathsf{...}</code>	<code>\$\mathsf{G\times R}\$</code>	$G \times R$
<code>\mathtt{...}</code>	<code>\$\mathtt{L}(b,c)\$</code>	$L(b, c)$
<code>\mathnormal{...}</code>	<code>\$\mathnormal{R_{19}}\backslash neq R_{19}\$</code>	$R_{19} \neq R_{19}$
<code>\mathit{...}</code>	<code>\$\mathit{ffi}\backslash neq ffi\$</code>	$ffi \neq ffi$

Л^AT_EX: разделению логической и визуальной разметки вашего документа. Это значит, что, если вы пользуетесь одними и теми же командами смены шрифта в разных местах для верстки специального вида информации, вы должны использовать `\newcommand` и определить команду, «оборачивающую» в себя команду смены шрифта.

```
% в преамбуле или пакете
\newcommand{\danger}[1]{\textbf{#1}}
% в документе
Не \danger{входите} в эту комнату.
Она занята \danger{машиной}
неизвестного назначения.
```

Не **входите** в эту комнату. Она занята **машиной** неизвестного назначения.

Этот подход имеет то преимущество, что вы позже можете решить, что хотите использовать другое визуальное представление опасности,¹ нежели `\textbf`, без необходимости пробираться через весь документ, отыскивая все вхождения `\textbf` и определяя, отмечает ли каждое из них опасность или что-нибудь другое.

6.2.3 Совет

Для завершения нашего путешествия в мир шрифтов и их размеров, позвольте дать вам один совет:

Помните! Чем БОЛЬШЕ шрифтов ВЫ используете в вашем документе, тем легче ЕГО ЧИТАТЬ и тем *красивее* он будет.

6.3 Интервалы

6.3.1 Интервалы между строками

Если вам нужны большие интервалы между строками, то их значение можно изменить помещением в преамбулу команды

```
\linespread{коэффициент}
```

Для печати «через полтора интервала» пользуйтесь `\linespread{1.3}`, для печати «через два интервала» — `\linespread{1.6}`. По умолчанию этот коэффициент равен 1.

Заметим, что эффект от команды `\linespread` довольно радикален, и поэтому она не подходит для публикуемых работ. Поэтому, если у вас есть весомые соображения для изменения межстрочного интервала, лучше пользуйтесь следующей

¹danger — Прим. перев.

командой:

```
\setlength{\baselineskip}{1.5\baselineskip}
```

```
{\setlength{\baselineskip}%
  {1.5\baselineskip}}
Этот абзац набран с интервалом в
1.5 раза больше предыдущего. Заметьте
команду \par{} в конце абзаца.\par}
```

Ясно, зачем набрат этот абзац: он показывает, что за закрывающейся фигурной скобкой все вернулось к нормальным установкам.

Этот абзац набран с интервалом в 1.5 раза больше предыдущего. Заметьте команду

в конце абзаца.

Ясно, зачем набрат этот абзац: он показывает, что за закрывающейся фигурной скобкой все вернулось к нормальным установкам.

6.3.2 Форматирование абзацев

Два параметра в L^AT_EX влияют на верстку абзацев. Поместив в преамбулу определения вида

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

вы измените внешний вид абзацев. Эти две строчки увеличивают расстояние между абзацами и устанавливают абзацный отступ равным нулю.

Части `plus` и `minus` говорят T_EX, что он может увеличивать и уменьшать интервал между абзацами на указанные величины, если это необходимо для правильного размещения абзацев на странице.

В Европе абзацы часто отделяют пробелами и не делают в них отступа. Однако, имейте в виду, что это влияет также и на оглавление: его строки тоже становятся теперь более разреженными. Чтобы этого избежать, эти команды можно перенести из преамбулы документа куда-нибудь после `\tableofcontents`, или не использовать их совсем, потому что в профессиональной книжной верстке используется выделение абзацев красной строкой, а не пробелами.

Если вы хотите сделать абзацный отступ в не имеющем его абзаце, то вставьте в начало абзаца команду¹

```
\indent
```

Понятно, что эффект от нее будет только если `\parindent` не установлен равным нулю.

¹Для добавления отступа к первому абзацу после каждого заголовка раздела пользуйтесь пакетом `indentfirst` из комплекта `'tools'`.

Таблица 6.5: Единицы размерности в $\text{\TeX}$

mm	миллиметр $\approx 1/25$ дюйма	
cm	сантиметр = 10 mm	
in	inch = 25.4 mm	
pt	пункт $\approx 1/72$ дюйма $\approx \frac{1}{3}$ mm	
em	примерная ширина буквы 'М' текущего шрифта	
ex	примерная высота буквы 'x' текущего шрифта	

Для создания абзаца без отступа первой командой абзаца можно сделать

\noindent

Это может быть удобно, когда вы начинаете документ с текста, а не с команды секционирования.

6.3.3 Горизонтальные интервалы

LaTeX автоматически определяет пробелы между словами и предложениями. Чтобы добавить горизонтальный пробел, пользуйтесь

\hspace{\textit{длина}}

Если такой интервал должен быть выдержан, даже если он приходится на начало или конец строки, используйте `\hspace*`, а не `\hspace`. В простейшем случае *длина* — это просто число и единица измерения. Наиболее важные единицы перечислены в таблице 6.5.

Тут\hspace{1.5cm}пробел
в 1,5см.

Тут пробел в 1,5см.

Команда

 $\backslash\mathrm{stretch}\{n\}$

генерирует специальный «резиновый» пробел. Он растягивается, заполняя все оставшееся места на строке. Если на одной строке встречаются две команды `\hspace{\stretch{n}}`, то они растягиваются пропорционально своим коэффициентам.

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

X X X

При использовании горизонтальных интервалов вместе с текстом может иметь смысл генерировать интервал, размер которого соотносится с размером текущего шрифта. Этого можно добиться при помощи относительных единиц размерности `em` и `ex`:

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



6.3.4 Вертикальные интервалы

Интервалы между абзацами, разделами, подразделами, ... определяются L^AT_EX автоматически. При необходимости дополнительный пробел *между двумя абзацами* можно добавить командой

```
\vspace{длина}
```

Обычно эта команда вставляется между двумя пустыми строчками. Если это пространство должно сохраняться вверху или внизу страницы, используйте вариант команды со звездочкой: `\vspace*`.

Команду `\stretch` вместе с `\pagebreak` можно применять для верстки текста на последней строке страницы или для вертикального центрирования текста на странице.

```
Некий текст\ldots
```

```
\vspace{\stretch{1}}
```

```
А это окажется на последней строке страницы.\pagebreak
```

Дополнительный пробел между двумя строками *одного* абзаца или внутри таблицы указывается командой

```
\\[длина]
```

При помощи `\bigskip` и `\smallskip` вы можете пропустить заранее определенные вертикальные интервалы, не задумываясь о конкретных числах.

6.4 Компоновка страницы

L^AT_EX 2_ε позволяет указать размер бумаги в команде `\documentclass`. Затем он автоматически выбирает правое поле. Но иногда предопределенные значения могут вас не устроить. Безусловно, вы их можете изменить. Иллюстрация 6.2 показывает

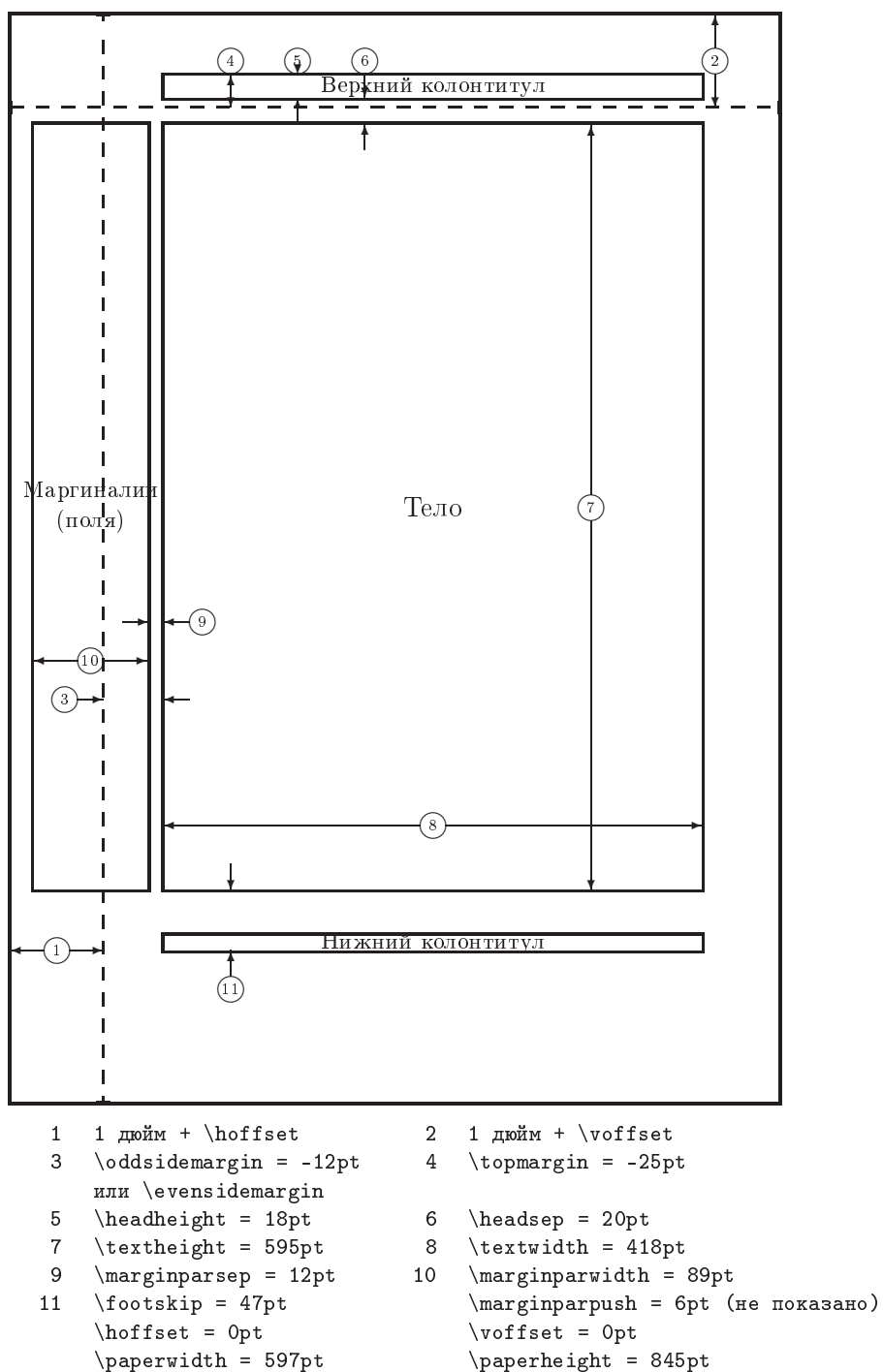


Рис. 6.2: Параметры компоновки страницы

все параметры, которые можно изменить. Она была сгенерирована пакетом `layout` из комплекта ‘tools’.¹

ПОДОЖДИТЕ! ... прежде, чем немедленно броситься делать эту слишком узкую страницу слегка пошире, потратьте несколько секунд на размышления. Подобно другим вещам, выбор компоновки страницы в $\text{\LaTeX}$ весьма продуман.

Безусловно, если сравнить со страницей, выданной свежеставленным MS Word, то страницы $\text{\LaTeX}$ выглядят ужасно узкими. Однако, взгляните на вашу любимую книгу² и посчитайте количество букв на одной строчке. Вы обнаружите, что на каждой строчке не больше 66 букв. Теперь повторите это со страницей $\text{\LaTeX}$. Вы увидите, что и здесь тоже около 66 букв в строке. Опыт показывает, что при большем количестве букв чтение затрудняется, потому, что глазам становится труднее переходить от конца одной строки к началу следующей. Именно поэтому газеты часто верстаются в несколько колонок.

Так что, увеличивая ширину вашего текста, имейте в виду, что вы затрудняете жизнь его читателям. Однако, достаточно предупреждений, вам был обещан рассказ о том, как же это сделать...

$\text{\LaTeX}$ предоставляет две команды для изменения этих параметров. Их обычно используют в преамбуле документа.

Первая команда присваивает фиксированное значение любому параметру:

```
\setlength{параметр}{длина}
```

Вторая команда прибавляет длину к любому параметру:

```
\addtolength{параметр}{длина}
```

Она даже более полезна, чем `\setlength`, потому что позволяет вам делать настройку относительно существующих установок. Чтобы добавить сантиметр к общей ширине текста, например, в преамбулу нужно поместить следующее:

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

Здесь вам может быть интересен пакет `calc`, который позволяет использовать арифметические операции в аргументе `\setlength` и в других местах, где аргументами являются численные значения.

6.5 Еще о длинах

Всегда, когда это возможно, избегайте использовать в документах абсолютных величин. Лучше основывайтесь на ширине или высоте других элементов страницы.

¹CTAN:/tex-archive/macros/latex/required/tools

²Имея в виду настоящую печатную книгу, выпущенную уважаемым издательством.

Для ширины иллюстрации этим может служить `\textwidth`, чтобы она заполняла страницу целиком.

Следующие три команды позволяют определить ширину, высоту и глубину текстовой строки.

```
\settoheight{переменная}{текст}
\settodepth{переменная}{текст}
\settowidth{переменная}{текст}
```

Нижеследующий пример показывает возможное применение этих команд.

```
\flushleft
\newenvironment{vardesc}[1]{%
  \settowidth{\parindent}{#1:\ }
  \makebox[0pt][r]{#1:\ }}{}
```

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

```
\begin{vardesc}{Где}$a$,
$b$ -- прилегают к прямому углу
прямоугольного треугольника.
```

```
$c$ -- одинокая гипотенуза
этого треугольника.
```

```
$d$ -- вообще тут не участвует.
Вот загадка\ldots
\end{vardesc}
```

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

Где: a , b – прилегают к прямому углу
прямоугольного треугольника.

c – одинокая гипотенуза этого треугольника.

d – вообще тут не участвует. Вот загадка. . .

6.6 Блоки

Л^AT_EX выстраивает страницы, передвигая блоки. Сначала каждая буква является маленьким блоком, который приклеивается к другим буквам, формируя слово. Слова склеиваются с другими словами, но специальным эластичным клеем, который может растягиваться или сжиматься, так, чтобы в точности заполнить строку.

Надо признать, что это довольно упрощенная версия того, что происходит на самом деле, но идея в том, что T_EX всегда работает с блоками и клеем. Не только буква может быть блоком. Вы можете поместить в блок практически все, что угодно, не исключая и другие блоки. Каждый блок затем обрабатывается Л^AT_EX, как если бы это была отдельная буква.

В предыдущих главах вы уже встречали некоторые блоки, хотя об этом и не говорилось. Примерами могут быть окружение `tabular` или `\includegraphics`, оба производящие блок. Это значит, что вы легко можете сверстать рядом две таблицы или иллюстрации. Только убедитесь, что их общая ширина не превышает `\textwidth`.

Вы также можете упаковать любой абзац в блок или командой

```
\parbox[поз]{ширина}{текст}
```

или окружением

```
\begin{minipage}[поз]{ширина} текст \end{minipage}
```

Параметр *поз* может принимать одну из букв **c**, **t** или **b**, контролируя вертикальное выравнивание блока по отношению к базовой линии окружающего текста. *Ширина* принимает аргументом длину, определяющую ширину блока. Основное отличие между `\minipage` и `\parbox` — в том, что внутри `\parbox` можно использовать не все команды и окружения, тогда как внутри `\minipage` можно практически все.

В то время, как `\parbox` упаковывает целый абзац, разбивая строки и прочее, существует класс блочных команд, работающих только на горизонтально расположенном материале. Одну из них мы уже знаем. Она называется `\mbox` и просто упаковывает последовательность блоков, что можно использовать для предотвращения переноса ЛАТЭХ двух слов. Так как вы можете помещать одни блоки в другие, эти упаковщики горизонтальных блоков чрезвычайно гибки.

```
\makebox[ширина][поз]{текст}
```

Ширина определяет ширину результирующего блока так, как его видно снаружи.¹ Кроме выражений длины, вы тут можете использовать `\width`, `\height`, `\depth` и `\totalheight`. Они устанавливаются равными значениям, полученным измерением параметров *текста*.² Параметр *поз* принимает однобуквенное значение: **c**: центрировать, **l**: отжать влево, **r**: отжать вправо или **s**: равномерно заполнить блок текстом.

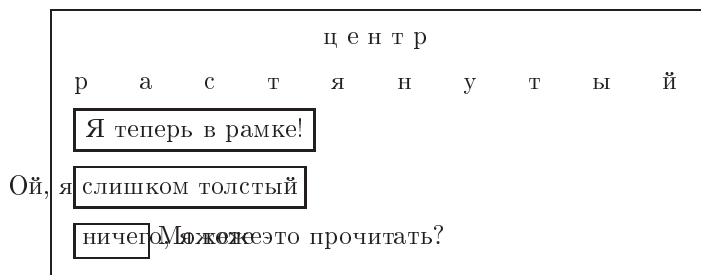
Команда `\framebox` работает в точности так же, как `\makebox`, но рисует рамку вокруг текста.

Следующий пример показывает некоторые возможности использования команд `\makebox` и `\framebox`.

¹Это означает, что она может быть меньше, чем материал внутри блока. В предельном случае вы можете даже установить ее в 0pt, так что текст внутри блока верстается, вообще не оказывая влияния на окружающие блоки.

²Ширина, высота, глубина и общая высота (высота плюс глубина) текста, соответственно. — Прим. перев.

```
\makebox[\textwidth]{%
  ц е н т р}\par
\makebox[\textwidth][s]{%
  р а с т я н у т ы й}\par
\framebox[1.1\width]{Я теперь
  в рамке!}\par
\framebox[0.8\width][r]{Ой,
  я слишком толстый}\par
\framebox[1cm][l]{ничего,
  я тоже}
Можете это прочитать?
```

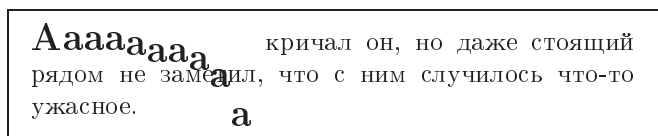


Теперь, когда мы управляем горизонталью, очевидный следующий шаг — вертикаль. Никаких проблем. Команда

```
\raisebox{сдвиг}[глубина][высота]{текст}
```

позволяет вам определить вертикальные характеристики блока. В первых трех параметрах можно использовать `\width`, `\height`, `\depth` и `\totalwidth`, чтобы получить размеры аргумента *текст*.

```
\raisebox{0pt}[0pt][0pt]{\Large%
\textbf{Aaaa}\raisebox{-0.3ex}{a}%
\raisebox{-0.7ex}{aa}%
\raisebox{-1.2ex}{a}%
\raisebox{-2.2ex}{a}%
\raisebox{-4.5ex}{a}}
кричал он, но даже стоящий рядом
не заметил, что с ним случилось
что-то ужасное.
```



6.7 Линейки и распорки

Несколько страниц назад вы могли отметить команду

```
\rule[сдвиг]{ширина}{высота}
```

При обычном использовании она генерирует простой черный блок.

```

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

```



Это можно использовать для рисования вертикальных и горизонтальных линий. Например, линия на титульном листе нарисована командой `\rule`.

Специальным случаем является линейка, у которой нет ширины, но есть определенная высота. В профессиональной верстке ее называют *распоркой*. Ее используют, чтобы обеспечить определенную минимальную высоту элемента страницы. Вы можете использовать ее, чтобы сделать строку окружения `tabular` имеющей определенную минимальную высоту.

```

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

```

Pitprop ...
Strut

Конец.

Литература

- [1] Leslie Lamport. *L^AT_EX: A Document Preparation System*. Addison-Wesley, Reading, Massachusetts, второе издание, 1994, ISBN 0-201-52983-1.
- [2] Donald E. Knuth. *The T_EXbook*, Volume A of *Computers and Typesetting*, Addison-Wesley, Reading, Massachusetts, second edition, 1984, ISBN 0-201-13448-9.
- [3] Michel Goossens, Frank Mittelbach and Alexander Samarin. *The L^AT_EX Companion*. Addison-Wesley, Reading, Massachusetts, 1994, ISBN 0-201-54199-8.¹
- [4] Michel Goossens, Sebastian Rahtz and Frank Mittelbach. *The L^AT_EX Graphics Companion*. Addison-Wesley, Reading, Massachusetts, 1997, ISBN 0-201-85469-4.²
- [5] Каждая установка L^AT_EX должна содержать так называемый *L^AT_EX Local Guide*, объясняющий особенности локальной системы. Он должен находиться в файле, называемом `local.tex`. К сожалению, некоторые ленивые администраторы такого документа не предоставляют. В таком случае просите о помощи местного L^AT_EX гуру.
- [6] L^AT_EX3 Project Team. *L^AT_EX 2_ε for authors*. Включен в поставку L^AT_EX 2_ε как `usrguide.tex`.
- [7] L^AT_EX3 Project Team. *L^AT_EX 2_ε for Class and Package writers*. Включен в поставку L^AT_EX 2_ε как `clsguide.tex`.
- [8] L^AT_EX3 Project Team. *L^AT_EX 2_ε Font selection*. Включен в поставку L^AT_EX 2_ε как `fntguide.tex`.
- [9] D. P. Carlisle. *Packages in the ‘graphics’ bundle*. Входит в состав комплекта ‘graphics’ как `grfguide.tex`, доступен оттуда же, откуда ваша поставка L^AT_EX.
- [10] Rainer Schöpf, Bernd Raichle, Chris Rowley. *A New Implementation of L^AT_EX’s verbatim Environments*. Входит в состав комплекта ‘tools’ как `verbatim.dtx`, доступен оттуда же, откуда ваша поставка L^AT_EX.
- [11] Vladimir Volovich, Werner Lemberg and L^AT_EX3 Project Team. *Cyrillic languages support in L^AT_EX*. Включен в поставку L^AT_EX 2_ε как `cyrguide.tex`.

¹Издан русский перевод: М.Гуссенс, Ф.Миттельбах, А.Самарин. *Путеводитель по пакету L^AT_EX и его расширению L^AT_EX 2_ε*. Мир, 1999, ISBN 5-03-003325-4.

²Издан русский перевод: М.Гуссенс, С.Ратц и Ф.Миттельбах. *Путеводитель по пакету L^AT_EX и его графическим расширениям*. Мир, 2002, ISBN 5-03-003388-2.

-
- [12] Graham Williams. *The TeX Catalogue* полный список множества пакетов, имеющих отношение к $\text{\TeX}$ и $\text{\LaTeX}$. Доступен в Интернет по адресу CTAN:/tex-archive/help/Catalogue/catalogue.html
 - [13] Keith Reckdahl. *Using EPS Graphics in $\text{\LaTeX} 2_{\epsilon}$ Documents* объясняет все, что вы когда бы то ни было хотели знать про EPS файлы и их использование в документах $\text{\LaTeX}$. Доступен в Интернет по адресу CTAN:/tex-archive/info/epslatex.ps
 - [14] Kristoffer H. Rose. *Xy-pic User's Guide*. Доступен в Интернет с CTAN вместе с дистрибутивом $\text{\Xy-pic}$.
 - [15] John D. Hobby. *A User's Manual for MetaPost*. Доступен по адресу <http://cm.bell-labs.com/who/hobby/>
 - [16] Alan Hoenig. *TeX Unbound*. Oxford University Press, 1998, ISBN 0-19-509685-1; 0-19-509686-X (pbk.)
 - [17] Urs Oswald. *Graphics in $\text{\LaTeX} 2_{\epsilon}$* , содержит набор исходных текстов на Java для генерации произвольных окружностей и эллипсов из окружения `picture`, и *MetaPost - A Tutorial*. Оба доступны по адресу <http://www.ursoswald.ch>

Предметный указатель

\!, 53
", 21
"', 32
"-, 32
"---, 32
"<, 32
"=, 32
">, 32
"‘, 32
\$, 47
, 50
\(, 47
\), 47
\., 48, 53
-, 22
—, 22
\-, 20
—, 22
—, 22
..., 23
\:, 53
\;, 53
«, 21
», 21
\@, 33
\[, 47
\[, 19, 38–40, 108
*, 19
\], 47
^, 50
_, 50
~, 33

Acrobat Reader, 74
\addtolength, 110
æ, 24
aeguill, 76

\Alph, 33
\alph, 33
amsbsy, 59
amsmath, 51–54, 56, 57, 59
amssymb, 49, 60
\and, 35
\appendix, 34, 35
\ar, 96
\arccos, 51
\arcsin, 51
\arctan, 51
\arg, 51
array, 54, 55
\Asbuk, 33
\asbuk, 33
\author, 35, 79

babel, 6, 20, 25, 28, 31–33, 83
\background, 83
\backmatter, 35
backslash, 5
\backslash, 4
\begin, 37, 86, 94
\bibitem, 69
\Big, 53
\big, 53
\Bigg, 53
\bigg, 53
\bigskip, 108
\binom, 51
bm, 59
\bmod, 51
\boldmath, 59
\boldsymbol, 59

calc, 110
\caption, 44, 45

- `\cdot`, 51
- `\cdots`, 53
 - center, 38
- `\chapter`, 34, 71
- `\chaptermark`, 71
- `\ci`, 99
- `\circle`, 89
- `\circle*`, 89
- `\cite`, 69
- `\cleardoublepage`, 45
- `\clearpage`, 45
- `\cline`, 40
 - color, 77, 81, 82
 - comment, 6
- `\cos`, 51
- `\cosh`, 51
- `\cot`, 51
- `\coth`, 51
- `\csc`, 51
- `\cyrmathrm`, 57
- `\date`, 35
 - dcolumn, 41
- `\ddots`, 53
- `\deg`, 51
- `\depth`, 112, 113
 - description, 37
- `\det`, 51
- `\dim`, 51
 - displaymath, 47
- `\displaystyle`, 57
 - doc, 12
- `\documentclass`, 9, 20, 75
- `\dq`, 29
- `\dum`, 99
 - eepic, 85, 89
- `\emblema`, 83
- `\emph`, 36, 103
 - empty, 13
 - Encapsulated POSTSCRIPT, 67, 77, 86
- `\end`, 37, 86
- `\enumBul`, 33
- `\enumEng`, 33
 - enumerate, 37
- `\enumLat`, 33
 - epic, 85
- `\eqnarray`, 55
- `\eqref`, 48
 - equation, 48, 55
- `\eufrak`, 66
- `\EUR`, 23
 - eurosym, 23
- `\euscript`, 66
- `\exp`, 51
 - exscale, 12, 53
- `fancyhdr`, 71, 72
- `\fbox`, 21
 - figure, 42, 44, 67
- `\flq`, 29
- `\flqq`, 21, 29
 - flushleft, 38
 - flushright, 38
- `foiltex`, 9
- `\foldera`, 93
- `\folderb`, 93
 - fontenc, 12, 31, 32
 - fontent, 26
- `\footnote`, 36, 45
- `\footnotesize`, 103
- `\frac`, 51
- `\framebox`, 112
- `\frenchspacing`, 32, 33
- `\frontmatter`, 35
- `\frq`, 29
- `\frqq`, 21, 29
- `\fussy`, 20
- `\gcd`, 51
 - geometry, 73
 - GhostScript, 67
- `\glqq`, 21
 - graphicx, 67, 68, 77, 81, 82
- `\grqq`, 21
- `HLATEX`, 30
- `hLATEXp`, 30
- headings, 13

- \height, 112, 113
- \hline, 40
- \hom, 51
- \href, 79, 81
- \hspace, 101, 107
- \Huge, 103
- \huge, 103
 - hyperref, 75, 77, 78, 82
 - hyphenat, 73
- \hyphenation, 20
 - и j без точек, 24
- \idotsint, 54
 - ifthen, 12
- \iiiint, 54
- \iiint, 54
- \iint, 54
- \include, 14
- \includegraphics, 68, 77, 81, 111
- \includeonly, 14
- \indent, 106
 - indentfirst, 106
- \index, 70, 71
- \inf, 51
- \input, 14
 - inputenc, 12, 25, 28, 31
- \int, 52
- \item, 37
 - itemize, 37
- \ker, 51
 - Knuth, Donald E., 1
 - koi8-r, 32
- \label, 35, 36, 48
 - Lamport, Leslie, 1
- \LARGE, 103
- \Large, 103
- \large, 103
- \LaTeX, 21
 - Л^AT_EX3, 3
- \LaTeXe, 21
 - latexsym, 12
 - layout, 110
- \ldots, 23, 53
- \left, 52
- \leftmark, 71
- \lg, 51
 - LGR, 26
- \lim, 51
- \liminf, 51
- \limsup, 51
- \line, 87, 93
- \linebreak, 19
- \linespread, 105
- \linethickness, 90, 91, 93
- \listoffigures, 44
- \listoftables, 44
- \ln, 51
- \log, 51
 - longtable, 42
 - lscommand, 99
- \mainmatter, 35, 80
- \makebox, 112
 - makeidx, 12, 70
- \makeindex, 70
- \maketitle, 35
- \marginwidth, 83
 - marvosym, 23
- \math, 47
- \mathbb, 49
- \mathbf, 104
- \mathcal, 104
- \mathit, 104
- \mathnormal, 104
- \mathrm, 57, 104
 - mathrsfs, 66
- \mathsf, 104
- \mathtext, 32
- \mathtt, 104
- \max, 51
- \mbox, 21, 24, 112
 - METAPOST, 77
- \min, 51
- \minipage, 112
 - minipage, 112
- Mittelbach, Frank, 1
- mltex, 76

- mltex, 76
- \multicolumn, 41
- \multirow, 86, 90
- \newcommand, 100
- \newenvironment, 101
- \newline, 19
- \newpage, 19
- \newsavebox, 92
- \newtheorem, 57, 58
- \noindent, 107
- \nolinebreak, 19
- \nonumber, 56
- \nopagebreak, 19
- \normalsize, 103
- \not, 61
- œ, 24
- OT1, 26
- \oval, 91, 93
- \overbrace, 50
 - overfull hbox, 19
- \overlay, 83
- \overleftarrow, 50
- \overline, 50
- \overrightarrow, 50
- \pagebreak, 19
- \pageref, 35, 74
- \pagestyle, 13
- \panelwidth, 83
- \paragraph, 34
- \parbox, 112
- \parindent, 106
- \parskip, 106
- \part, 34
 - pause, 83, 84
- \pause, 83
 - PDF, 74
 - pdfL^AT_EX, 75, 82
 - pdfscreen, 82–84
 - pdfL^AT_EX, 74
 - pdfT_EX, 74
- \phantom, 45, 56
- picture, 85, 86, 89, 90
 - plain, 13
- \pmod, 51
 - POSTSCRIPT, 3, 8, 31, 67, 68, 74–76, 85
 - Encapsulated, 67, 77, 86
- \Pr, 51
- \printindex, 71
- \prod, 52
- \protect, 45
- \providecommand, 100
- \ProvidesPackage, 101
 - pspicture, 85
 - pstricks, 85, 89
- \put, 86–92
 - pxfonts, 76
- \qbezier, 85, 86, 93
- \qqquad, 48, 53
- \quad, 48, 53
 - quotation, 39
 - quote, 38
- \raisebox, 113
- \ref, 35, 48, 74
- \renewcommand, 100
- \renewenvironment, 101
- \right, 52
- \right., 53
- \rightmark, 71
- \rule, 101, 113, 114
- \savebox, 92
- \screensize, 83
- \scriptscriptstyle, 57
- \scriptsize, 103
- \scriptstyle, 57
- \sec, 51
- \section, 34, 45, 71
- \sectionmark, 71
- \selectlanguage, 25
- \setlength, 86, 106, 110
- \settodepth, 111
- \settoheight, 111
- \settowidth, 111
 - showidx, 71
- \sin, 51

- \sinh, 51
- \sloppy, 20
- \small, 103
- \smallskip, 108
- \sqrt, 50
- \stackrel, 52
- \stretch, 101, 107
 - subarray, 52
- \subparagraph, 34
- \subsection, 34
- \subsectionmark, 71
- \substack, 52
- \subsubsection, 34
- \sum, 52
- \sup, 51
 - supertabular, 42
 - syntonly, 12, 15
- T1, 26, 32
- T2*, 31
- T2A, 26, 32
- T2B, 26
- T2C, 26
- table, 42, 44, 67
- \tableofcontents, 34
 - tabular, 40, 54, 111
- \tan, 51
- \tanh, 51
- \TeX, 21
- \texorpdfstring, 80
- \textbf, 103
 - textcomp, 22
- \texteuro, 22
- \textit, 103
- \textmd, 103
- \textnormal, 103
- \textrm, 57, 103
- \textsc, 103
- \textsf, 103
- \textsl, 103
- \textstyle, 57
- \texttt, 103
- \textup, 103
 - thebibliography, 69
- \thicklines, 88, 91, 93
- \thinlines, 91, 93
- \thispagestyle, 13
- \tiny, 103
- \title, 35
- \tnss, 100
- \today, 21
- \totalheight, 112
- \totalwidth, 113
 - txfonts, 76
- ucs, 26
- \underbrace, 50
 - underfull hbox, 20
- \underline, 36, 50
- \unitlength, 86, 87
- URL, 22
- \usebox, 92
- \usepackage, 11, 22, 23, 25, 26, 101
 - utf8, 26
- \vdots, 53
- \vec, 50
- \vector, 88
- \verb, 39, 40
 - verbatim, 6, 72
 - verbatim, 39, 72
- \verbatim@font, 72
- \verbatiminput, 72
 - verse, 39
- \vspace, 108
- \widehat, 50
- \widetilde, 50
- \width, 112, 113
 - www, 22
 - WYSIWYG, 2
- X2, 26
- xpdf, 74
- xy, 95
- \xymatrix, 96
- ажурные полужирные символы, 49
- акцент

- математический, 50
- акценты, 24
 - acute, 24
 - grave, 24
 - umlaut, 24
- Болгарский, 31
- Бразильский, 27
- без засечек, 103
- библиография, 69
- буквы
 - европейские, 24
- векторы, 50
- верхние индексы, 50
- входная кодировка
 - koi8-r, 32
 - utf8, 26
- входной файл, 6
- выделение, 36
- выравнивание
 - вправо или влево, 38
 - по десятичной точке, 41
- гипертекст, 74
- графика, 9, 67
- греческие буквы, 49
- группирование, 102
- два столбца, 10
- двусторонний вывод, 10
- дефис, 22
- длина, 107
- длинное тире, 22
- длинные уравнения, 55
- дробь, 51
- другие языки, 24
- единицы, 107
- заголовок документа, 10
- запятая, 23
- знак градуса, 22
- знак минуса, 22
- иллюстрации, 43
- интервал
 - двойной, 105
 - междустрочный, 105
- Корейский, 29
- кавычки, 21
- капиталь, 103
- квадратные скобки, 5
- квадратный корень, 50
- класс
 - article, 9
 - book, 9
 - report, 9
 - slides, 9
- кодировка
 - входная
 - koi8-r, 32
 - utf8, 26
 - шрифта
 - LGR, 26
 - OT1, 26
 - T1, 26, 32
 - T2*, 31
 - T2A, 26, 32
 - T2B, 26
 - T2C, 26
 - X2, 26
- кодировка шрифта, 12, 26
 - LGR, 26
 - OT1, 26
 - T1, 26, 32
 - T2*, 31
 - T2A, 26, 32
 - T2B, 26
 - T2C, 26
 - X2, 26
- колонтитул
 - верхний, 13
 - нижний, 13
- команда
 - \!, 53
 - \(, 47

\), 47
\,, 48, 53
\-, 20
\:, 53
\;, 53
\@, 33
\[, 47
\, 19, 38–40, 108
*, 19
\], 47
\addtolength, 110
\Alph, 33
\alph, 33
\and, 35
\appendix, 34, 35
\ar, 96
\arccos, 51
\arcsin, 51
\arctan, 51
\arg, 51
\Asbuk, 33
\asbuk, 33
\author, 35, 79
\background, 83
\backmatter, 35
\backslash, 4
\begin, 37, 86, 94
\bibitem, 69
\Big, 53
\big, 53
\Bigg, 53
\bigg, 53
\bigskip, 108
\binom, 51
\bmod, 51
\boldmath, 59
\boldsymbol, 59
\caption, 44, 45
\cdot, 51
\cdots, 53
\chapter, 34, 71
\chaptermark, 71
\ci, 99
\circle, 89
\circle*, 89
\cite, 69
\cleardoublepage, 45
\clearpage, 45
\cline, 40
\cos, 51
\cosh, 51
\cot, 51
\coth, 51
\csc, 51
\cyrmathrm, 57
\date, 35
\ddots, 53
\deg, 51
\depth, 112, 113
\det, 51
\dim, 51
\displaystyle, 57
\documentclass, 9, 20, 75
\dq, 29
\dum, 99
\emblema, 83
\emph, 36, 103
\end, 37, 86
\enumBul, 33
\enumEng, 33
\enumLat, 33
\eqref, 48
\EUR, 23
\exp, 51
\fbox, 21
\flq, 29
\flqq, 21, 29
\foldera, 93
\folderb, 93
\footnote, 36, 45
\frac, 51
\framebox, 112
\frenchspacing, 32, 33
\frontmatter, 35
\frq, 29
\frqq, 21, 29
\fussy, 20
\gcd, 51

<code>\glqq</code> , 21	<code>\marginwidth</code> , 83
<code>\grqq</code> , 21	<code>\mathbb</code> , 49
<code>\height</code> , 112, 113	<code>\mathrm</code> , 57
<code>\hline</code> , 40	<code>\max</code> , 51
<code>\hom</code> , 51	<code>\mbox</code> , 21, 24, 112
<code>\href</code> , 79, 81	<code>\min</code> , 51
<code>\hspace</code> , 101, 107	<code>\minipage</code> , 112
<code>\hyphenation</code> , 20	<code>\multicolumn</code> , 41
<code>\idotsint</code> , 54	<code>\multispan</code> , 86, 90
<code>\iiiint</code> , 54	<code>\newcommand</code> , 100
<code>\iiint</code> , 54	<code>\newenvironment</code> , 101
<code>\iint</code> , 54	<code>\newline</code> , 19
<code>\include</code> , 14	<code>\newpage</code> , 19
<code>\includegraphics</code> , 68, 77, 81, 111	<code>\newsavebox</code> , 92
<code>\includeonly</code> , 14	<code>\newtheorem</code> , 57, 58
<code>\indent</code> , 106	<code>\noindent</code> , 107
<code>\index</code> , 70, 71	<code>\nolinebreak</code> , 19
<code>\inf</code> , 51	<code>\nonumber</code> , 56
<code>\input</code> , 14	<code>\nopagebreak</code> , 19
<code>\int</code> , 52	<code>\not</code> , 61
<code>\item</code> , 37	<code>\oval</code> , 91, 93
<code>\ker</code> , 51	<code>\overbrace</code> , 50
<code>\label</code> , 35, 36, 48	<code>\overlay</code> , 83
<code>\LaTeX</code> , 21	<code>\overleftarrow</code> , 50
<code>\LaTeXe</code> , 21	<code>\overline</code> , 50
<code>\ldots</code> , 23, 53	<code>\overrightarrow</code> , 50
<code>\left</code> , 52	<code>\pagebreak</code> , 19
<code>\leftmark</code> , 71	<code>\pageref</code> , 35, 74
<code>\lg</code> , 51	<code>\pagestyle</code> , 13
<code>\lim</code> , 51	<code>\panelwidth</code> , 83
<code>\liminf</code> , 51	<code>\paragraph</code> , 34
<code>\limsup</code> , 51	<code>\parbox</code> , 112
<code>\line</code> , 87, 93	<code>\parindent</code> , 106
<code>\linebreak</code> , 19	<code>\parskip</code> , 106
<code>\linespread</code> , 105	<code>\part</code> , 34
<code>\linethickness</code> , 90, 91, 93	<code>\pause</code> , 83
<code>\listoffigures</code> , 44	<code>\phantom</code> , 45, 56
<code>\listoftables</code> , 44	<code>\pmod</code> , 51
<code>\ln</code> , 51	<code>\Pr</code> , 51
<code>\log</code> , 51	<code>\printindex</code> , 71
<code>\mainmatter</code> , 35, 80	<code>\prod</code> , 52
<code>\makebox</code> , 112	<code>\protect</code> , 45
<code>\makeindex</code> , 70	<code>\providecommand</code> , 100
<code>\maketitle</code> , 35	<code>\ProvidesPackage</code> , 101

- `\put`, 86–92
- `\qbezier`, 85, 86, 93
- `\qqquad`, 48, 53
- `\quad`, 48, 53
- `\raisebox`, 113
- `\ref`, 35, 48, 74
- `\renewcommand`, 100
- `\renewenvironment`, 101
- `\right`, 52
- `\right.`, 53
- `\rightmark`, 71
- `\rule`, 101, 113, 114
- `\savebox`, 92
- `\screensize`, 83
- `\scriptscriptstyle`, 57
- `\scriptstyle`, 57
- `\sec`, 51
- `\section`, 34, 45, 71
- `\sectionmark`, 71
- `\selectlanguage`, 25
- `\setlength`, 86, 106, 110
- `\settodepth`, 111
- `\settoheight`, 111
- `\settowidth`, 111
- `\sin`, 51
- `\sinh`, 51
- `\sloppy`, 20
- `\smallskip`, 108
- `\sqrt`, 50
- `\stackrel`, 52
- `\stretch`, 101, 107
- `\subparagraph`, 34
- `\subsection`, 34
- `\subsectionmark`, 71
- `\substack`, 52
- `\subsubsection`, 34
- `\sum`, 52
- `\sup`, 51
- `\tableofcontents`, 34
- `\tan`, 51
- `\tanh`, 51
- `\TeX`, 21
- `\texorpdfstring`, 80
- `\texteuro`, 22
- `\textrm`, 57
- `\textstyle`, 57
- `\thicklines`, 88, 91, 93
- `\thinlines`, 91, 93
- `\thispagestyle`, 13
- `\title`, 35
- `\tnss`, 100
- `\today`, 21
- `\totalheight`, 112
- `\totalwidth`, 113
- `\underbrace`, 50
- `\underline`, 36, 50
- `\unitlength`, 86, 87
- `\usebox`, 92
- `\usepackage`, 11, 22, 23, 25, 26, 101
- `\vdots`, 53
- `\vec`, 50
- `\vector`, 88
- `\verb`, 39, 40
- `\verbatim@font`, 72
- `\verbatiminput`, 72
- `\vspace`, 108
- `\widehat`, 50
- `\widetilde`, 50
- `\width`, 112, 113
- `\xymatrix`, 96
- команды, 5
 - хрупкие, 45
- комментарии, 5
- компоновка страницы, 108
- корейский шрифт
 - УНС шрифт, 31
- короткое тире, 22
- курсив, 103
- лигатура, 24
- линии
 - горизонтальные, 50
- лист
 - A4, 10
 - A5, 10
 - B5, 10
 - executive, 10
 - legal, 10

- letter, 10
- титульный, 10, 35
- математика, 47
- математические
 - функции, 51
- математический
 - минус, 22
 - ограничитель, 53
 - пробел, 53
- многоточие, 23
- Немецкий, 28
- наклонный шрифт, 103
- немецкий язык, 25
- необязательные параметры, 5
- нижние индексы, 50
- оглавление, 34
- ограничители, 52
- один столбец, 10
- односторонний вывод, 10
- окружение, 37
 - array, 54, 55
 - center, 38
 - comment, 6
 - description, 37
 - displaymath, 47
 - enumerate, 37
 - eqnarray, 55
 - equation, 48, 55
 - figure, 42, 44, 67
 - flushleft, 38
 - flushright, 38
 - itemize, 37
 - lscommand, 99
 - math, 47
 - minipage, 112
 - picture, 85, 86, 89, 90
 - pspicture, 85
 - quotation, 39
 - quote, 38
 - subarray, 52
 - table, 42, 44, 67
 - tabular, 40, 54, 111
 - thebibliography, 69
 - verbatim, 39, 72
 - verse, 39
- оператор
 - интеграла, 52
 - произведения, 52
 - суммы, 52
- опции, 9
- Португальский, 27
- пакет, 6, 9, 99
 - aeguill, 76
 - ambsy, 59
 - amsfonts, 49, 66
 - amsmath, 51–54, 56, 57, 59
 - amssymb, 49, 60
 - babel, 6, 20, 25, 28, 31–33, 83
 - bm, 59
 - calc, 110
 - color, 77, 81, 82
 - dcolum, 41
 - doc, 12
 - eepic, 85, 89
 - epic, 85
 - eufrak, 66
 - eurosym, 23
 - euscript, 66
 - exscale, 12, 53
 - fancyhdr, 71, 72
 - fontenc, 12, 31, 32
 - fontent, 26
 - geometry, 73
 - graphicx, 67, 68, 77, 81, 82
 - hyperref, 75, 77, 78, 82
 - hyphenat, 73
 - ifthen, 12
 - indentfirst, 106
 - inputenc, 12, 25, 28, 31
 - latexsym, 12
 - layout, 110
 - longtable, 42
 - makeidx, 12, 70

- marvosym, 23
- mathrsfs, 66
- mathtext, 32
- mltex, 76
- pause, 83, 84
- pdfscreen, 82–84
- pstricks, 85, 89
- pxfonts, 76
- showidx, 71
- supertabular, 42
- syntonly, 12, 15
- textcomp, 22
- txfonts, 76
- ucs, 26
- verbatim, 6, 72
- ху, 95
- пакет CJK, 30
- пакет makeidx, 70
- параметр, 5
- перекрестные ссылки, 35
- плавающие объекты, 42
- поле, 108
- полужирные символы, 49, 59
- полужирный, 103
- правила переноса, 25
- преамбула, 6
- предметный указатель, 70
- преимущества L^AT_EX, 2
- пробел, 4
 - в начале строки, 4
 - вертикальный, 108
 - горизонтальный, 107
 - после команды, 5
- программа makeindex, 70
- производная, 50
- прямой шрифт, 103
- пустые символы, 4
- Русский, 31
- размер бумаги, 75
- размер бумаги, 108
- размер листа, 10
- размер основного шрифта, 10
- разрывы строк, 19
- распорка, 114
- расширение, 13
 - .aux, 14
 - .dvi, 13, 68
 - .eps, 68
 - .idx, 14
 - .ilg, 14
 - .ind, 14, 71
 - .lof, 14
 - .log, 13
 - .lot, 14
 - .sty, 73
 - .tex, 8
 - .toc, 14
- Слово, 71
- символы
 - зарезервированные, 4
- системы уравнений, 55
- скобки, 52
- специальные символы, 24
- спецификация размещения, 43
- стили страницы, 13
- стиль страницы
 - empty, 13
 - headings, 13
 - plain, 13
- стрелки, 50
- структура файла, 6
- таблицы, 43
- тильда, 22, 50
- тильда (~), 33
- типы файлов, 13
- тире, 22
 - длинное, 22
 - короткое, 22
- титульный лист, 35
- точка, 23
- точка, пробел после, 33
- точки
 - вертикальные, 53
 - горизонтальные, 53

- диагональные, 53
- три точки, 53
- Украинский, 31
- Французский, 27
- фигурные скобки, 5, 102
 - горизонтальные, 50
- формулы, 47
- функция модуля, 51
- цветной текст, 9
- шрифт, 102
 - `\footnotesize`, 103
 - `\Huge`, 103
 - `\huge`, 103
 - `\LARGE`, 103
 - `\Large`, 103
 - `\large`, 103
 - `\mathbf`, 104
 - `\mathcal`, 104
 - `\mathit`, 104
 - `\mathnormal`, 104
 - `\mathrm`, 104
 - `\mathsf`, 104
 - `\mathtt`, 104
 - `\normalsize`, 103
 - `\scriptsize`, 103
 - `\small`, 103
 - `\textbf`, 103
 - `\textit`, 103
 - `\textmd`, 103
 - `\textnormal`, 103
 - `\textrm`, 103
 - `\textsc`, 103
 - `\textsf`, 103
 - `\textsl`, 103
 - `\texttt`, 103
 - `\textup`, 103
 - `\tiny`, 103
- документа, размер, 10
- математический, размер, 56
- размер, 102, 103

