

Не очень краткое введение в L^AT_EX 2_ε

Или L^AT_EX 2_ε за 78 минут

by Tobias Oetiker

Hubert Partl, Irene Hyna and Elisabeth Schlegl

Version 3.2, 21. September, 1998

Перевод: Б. Тоботрас, 7 октября 1998 г.

Copyright ©1998 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 L^AT_EX 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 <no email>
in Graz

Если Вы интересуетесь немецким вариантом документа, вы можете найти его версию, которую Jörg Knappen обновил для L^AT_EX 2_ε, по адресу `CTAN:info/lkurz`.

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, David Carlisle, Christopher Chin, Chris McCormack,
Wim van Dam, David Dureisseix, Elliot, David Frey, Robin Fairbairns,
Alexandre Guimond, Cyril Goutte, Greg Gamble, Neil Hammond,
Rasmus Borup Hansen, Martien Hulsen, Werner Icking, Jakob, Eric Jacoboni,
Alan Jeffrey, Byron Jones, David Jones, Johannes-Maria Kaltenbach,
Andrzej Kawalec, Alain Kessi, Christian Kern, Jörg Knappen, Kjetil Kjernsmo,
Maik Lehradt, Martin Maechler, Claus Malten, Hubert Partl, John Reffling,
Mike Ressler, Brian Ripley, Young U. Ryu, Chris Rowley, Hanspeter Schmid,
Craig Schlenter, Josef Tkadlec, Didier Verna, Fabian Wernli, Fritz Zaucker,
Rick Zaccone, and 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 сверстаных 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].

L^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 L^AT_EX, должна быть предоставлена в *Local Guide* [4]. 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 L^AT_EX, а в том, чтобы научить, как писать ваши документы так, чтобы они могли быть обработаны L^AT_EX.

Это Введение разбито на четыре главы:

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 L^AT_EX. Оно поможет вам объединить в единое целое сведения из остальных глав.

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 сильных, сторон 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, включение EPS графики, и про некоторые другие полезные расширения.

Глава 5 содержит потенциально опасные сведения о том, как менять стандартный макет документа. Она расскажет вам, как легко испортить красивый вывод L^AT_EX.

Важно читать главы последовательно. В конце концов, книжка не такая уж большая. По мере чтения вам встретится большое количество примеров. Читайте их внимательно.

Если вам понадобится любой, относящийся к L^AT_EX, материал, поищите его на одном из ftp архивов CTAN. Для США это — `ctan.tug.org`, для Германии — `ftp.dante.de`, для Великобритании — `ftp.tex.ac.uk`. Если вы не в одной из этих стран, выберите ближайший к вам архив.

Если вы хотите иметь L^AT_EX на вашем собственном компьютере, посмотрите на то, что что доступно по адресу `CTAN: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 L^AT_EX на тему того, какие части введения легко понимаемы, и что можно объяснить лучше.

Tobias Oetiker <oetiker@ee.ethz.ch>

Department of Electrical Engineering,
Swiss Federal Institute of Technology

Текущая версия этого документа доступна на
`CTAN:info/lshort`¹.

¹Текущая версия перевода доступна на `http://xtalk.price.ru/tex`, а также на CTAN.

Оглавление

Благодарности	iii
Предисловие	iv
1. Это нужно знать	1
1.1. Названия	1
1.1.1. T _E X	1
1.1.2. L ^A T _E X	1
1.2. Основы	2
1.2.1. Автор, дизайнер и верстальщик	2
1.2.2. Дизайн макета	3
1.2.3. Преимущества и недостатки	3
1.3. Исходные файлы L ^A T _E X	4
1.3.1. Пробелы	4
1.3.2. Спецсимволы	5
1.3.3. Команды L ^A T _E X	5
1.3.4. Комментарии	6
1.4. Структура входного файла	6
1.5. Макеты документов	8
1.5.1. Классы документов	8
1.5.2. Пакеты	8
1.5.3. Стили страницы	11
1.6. Большие проекты	11
2. Верстка текста	13
2.1. Структура текста и языка	13
2.2. Разбиение на строки и страницы	15
2.2.1. Выровненные абзацы	15
2.2.2. Переносы	16
2.3. Специальные буквы и символы	16
2.3.1. Знаки кавычек	16
2.3.2. Тире и дефисы	17
2.3.3. Многоточие (...)	17
2.3.4. Лигатуры	17
2.3.5. Акценты и специальные символы	18
2.4. Поддержка иностранных языков	18

2.5. Пробелы между словами	19
2.6. Заголовки, главы и разделы	20
2.7. Перекрестные ссылки	21
2.8. Сноски	22
2.9. Выделенные слова	22
2.10. Окружения	23
2.10.1. Список, перечисление и описание	23
2.10.2. Выравнивание влево, вправо и по центру	24
2.10.3. Цитаты и стихи	24
2.10.4. Буквальное воспроизведение	25
2.10.5. Таблицы	26
2.11. Плавающие объекты	28
3. Набор математических формул	31
3.1. Общие сведения	31
3.2. Группировка в математическом режиме	33
3.3. Составляющие математической формулы	33
3.4. Математические пробелы	36
3.5. Вертикально расположенный материал	37
3.6. Размер математического шрифта	38
3.7. Теоремы, законы,	39
3.8. Полу жирные символы	40
3.9. Список математических символов	42
4. Специальные возможности	49
4.1. Включение EPS графики	49
4.2. Библиография	51
4.3. Указатели	52
4.4. Настраиваемые колонтитулы	53
4.5. Пакет verbatim	54
5. Настройка L^AT_EX	55
5.1. Новые команды, окружения и пакеты	55
5.1.1. Новые команды	56
5.1.2. Новые окружения	56
5.1.3. Ваш собственный пакет	57
5.2. Шрифты и их размеры	58
5.2.1. Команды смены шрифта	58
5.2.2. Опасность!	60
5.2.3. Совет	60
5.3. Интервалы	61
5.3.1. Интервалы между строками	61
5.3.2. Форматирование абзацев	61
5.3.3. Горизонтальные интервалы	61
5.3.4. Вертикальные интервалы	62
5.4. Компоновка страницы	63
5.5. Еще о длинах	65

5.6. Блоки	65
5.7. Линейки и распорки	67
Литература	69

Список рисунков

1.1. Составляющее \TeX системы	2
1.2. Минимальный файл \LaTeX	7
1.3. Пример реалистичной журнальной статьи	7
4.1. Пример настройки <code>fancyhdr</code>	53
5.1. Пример пакета	57
5.2. Параметры компоновки страницы	64

Список таблиц

1.1. Классы документов	8
1.2. Опции классов документов	9
1.3. Некоторые из распространяемых с L ^A T _E X пакетов	10
1.4. Предопределенные стили страницы L ^A T _E X	11
2.1. Акценты и специальные символы	18
2.2. Ключи размещения плавающего объекта	28
3.1. Акценты математического режима	42
3.2. Строчные греческие буквы	42
3.3. Прописные греческие буквы	42
3.4. Бинарные отношения	43
3.5. Бинарные операторы	43
3.6. Большие операторы	44
3.7. Стрелки	44
3.8. Ограничители	44
3.9. Большие ограничители	44
3.10. Прочие символы	45
3.11. Не-математические символы	45
3.12. Ограничители AMS	45
3.13. Буквы греческого и иврита AMS	45
3.14. Бинарные отношения AMS	46
3.15. Стрелки AMS	46
3.16. Отрицательные бинарные отношения и стрелки AMS	47
3.17. Бинарные операторы AMS	47
3.18. Прочие символы AMS	48
3.19. Математические алфавиты	48
4.1. Названия ключей для пакета <code>graphicx</code>	50
4.2. Примеры синтаксиса ключей указателя	52
5.1. Шрифты	58
5.2. Размеры шрифта	59
5.3. Абсолютные размеры шрифтов в стандартных классах	59
5.4. Математические шрифты	59
5.5. Единицы размерности в T _E X	62

Глава 1.

Это нужно знать

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

1.1. Названия

1.1.1. \TeX

\TeX — это компьютерная программа, созданная Дональдом Кнутом (Donald E. Knuth) [2]. Она предназначена для верстки текста и математических формул. Кнут начал писать \TeX в 1977 году из-за расстройства от того, что Американское Математическое Общество делало с его статьями в процессе их публикации. Где-то в 1974 году он даже прекратил посылать статьи: «просто мне было слишком больно смотреть на конечный результат». \TeX , в том виде, в каком мы его сегодня используем, был выпущен в 1982 году и слегка улучшен с годами. Последние несколько лет \TeX стал чрезвычайно стабилен. Кнут утверждает, что в нем практически нет ошибок. Номер версии \TeX сходится к π и сейчас равен 3.14159.

\TeX произносится как «тех». В среде ASCII \TeX нужно писать как `TeX`.

1.1.2. \LaTeX

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

В 1994 году пакет \LaTeX был обновлен командой $\LaTeX_{\text{X}}3$ во главе с Frank Mittelbach. В нем были сделаны некоторые давно ожидавшиеся улучшения, и вновь объединены все варианты \LaTeX , разошедшиеся с выпуска много лет назад версии \LaTeX 2.09. Чтобы не путать эту новую версию со старой, она называется $\LaTeX_{\text{X}}2_{\epsilon}$. Эта документация описывает именно $\LaTeX_{\text{X}}2_{\epsilon}$.

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

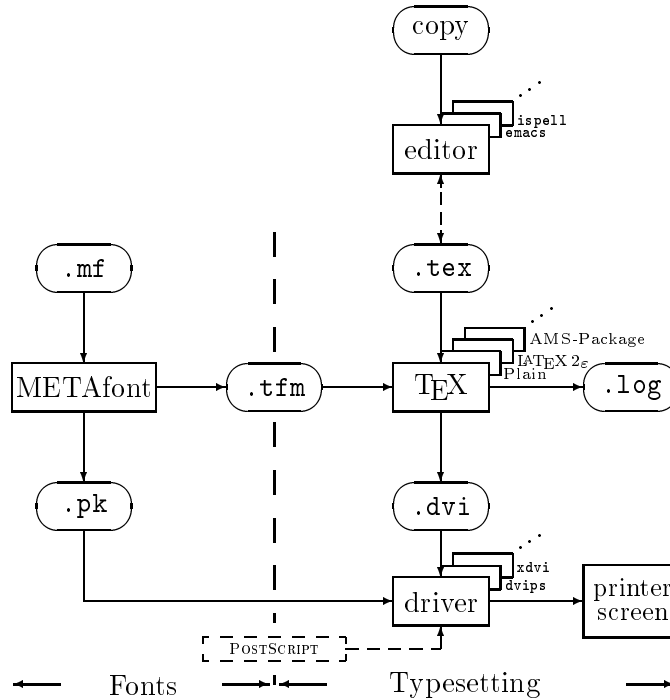
Рис. 1.1. Составляющее T_EX системы

Рисунок 1.1 на странице 2 показывает, как работают вместе T_EX и L^AT_EX 2_ε. Он взят из `wots.tex` (автор Kees van der Laan).

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 столбцов, 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 L^AT_EX, L^AT_EX берет на себя роль дизайнера книги, используя T_EX в качестве верстальщика. Но L^AT_EX — 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 L^AT_EX».

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 L^AT_EX обычно невозможно увидеть итоговую картину во время печатания текста. Ее, однако, можно посмотреть на экране после обработки файла L^AT_EX. 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. L^AT_EX предотвращает такие ошибки форматирования, заставляя автора объявлять *логическую* структуру его документа. Затем уже L^AT_EX выбирает наиболее подходящий макет (раскладку) документа.

1.2.3. Преимущества и недостатки

Тема, часто обсуждаемая, когда люди из мира WYSIWYG встречаются с пользователями L^AT_EX, — «преимущества L^AT_EX перед нормальными текстовыми процессорами», 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 L^AT_EX перед обычными текстовыми процессорами:

- Готовые профессионально выполненные макеты, делающие документы действительно выглядящими «как изданные».
- Удобно поддержана 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.

¹What you see is what you get.

- Для решения многих типографских задач, не поддерживаемых напрямую базовым \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 \LaTeX Companion* [3].
- \LaTeX поощряет авторов писать хорошо структурированные документы, так как именно так \LaTeX и работает — путем спецификации структуры.
- `\TeX`, формирующее сердце \LaTeX 2_ε, чрезвычайно мобилен и свободно доступен. Поэтому система работает практически на всех существующих платформах.

\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.

1.3. Исходные файлы \LaTeX

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

1.3.1. Пробелы

«Пустые» символы, такие, как пробел или табуляция, трактуются \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.	Неважно, вставляете ли вы один или несколько пробелов между словами.
Пустая строчка начинает новый абзац.	Пустая строчка начинает новый абзац.

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

1.3.2. Спецсимволы

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

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

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

$$\$ \& \% \# _ \{ \} \quad | \quad \backslash \$ \backslash \& \backslash \% \backslash \# \backslash _ \backslash \{ \backslash \}$$

Прочие символы, как и многие, многие другие, можно набрать специальными командами в математических формулах или как акценты. Знак « \backslash » *нельзя* вводить, добавляя перед ним еще один, так как эта команда ($\backslash\backslash$) используется для разрыва строки¹.

1.3.3. Команды \LaTeX

Команды \LaTeX чувствительны к регистру и принимают одну из следующих двух форм:

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

\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 \LaTeX игнорировать все пробелы после имени команды.

Я слышал, что Кнут разделяет людей, работающих с \TeX на \TeX ников и \TeX пертов. Сегодня — January 19, 1999	Я слышал, что Кнут разделяет людей, работающих с $\backslash\TeX\{ \}$ на $\backslash\TeX\{ \}$ ников и $\backslash\TeX$ пертов. $\backslash\backslash$ Сегодня~--- $\backslash\text{today}$
--	---

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 \LaTeX . Не задумывайтесь над ними, они будут разъяснены позже.

Вы можете <i>положиться</i> на меня!	Вы можете $\backslash\text{textsl}\{положиться\}$ на меня!
--------------------------------------	--

Пожалуйста, начните новую строку прямо тут! Спасибо!	Пожалуйста, начните новую строку прямо тут! $\backslash\text{newline}$ Спасибо!
---	--

¹Вместо этого пользуйтесь командой $\backslash\backslash\backslash\$$. Она дает « \backslash ».

1.3.4. Комментарии

Когда в процессе обработки входного файла \LaTeX встречается символ `%`, он игнорирует остаток текущей строки, возврат каретки и все пробелы в начале следующей строки. Это полезно для добавления в исходный файл замечаний, которые не будут выводиться на печать.

Это Spercalf <code>ifragilisticexpialidocious</code>	Это Spercalf <code>%</code> <code>ifragilistic%</code> <code>icexpialidocious</code>
--	--

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

1.4. Структура входного файла

Когда $\LaTeX 2_\epsilon$ 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 \LaTeX . Для загрузки такого пакета используется команда

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

Когда вся настройка закончена¹, вы начинаете тело текста командой

```
\begin{document}
```

Теперь вы вводите текст с командами \LaTeX . В конце документа вы добавляете команду

```
\end{document}
```

Все, что следует после нее, \LaTeX игнорирует.

Рис. 1.2 показывает содержимое минимального файла для $\LaTeX 2_\epsilon$. Несколько более сложный входной файл дан на рис. 1.3².

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

²На самом деле, для того, чтобы документы могли содержать русские буквы, необходимо подключить пакет русификации \LaTeX . Поскольку в настоящее время таких пакетов несколько, используемая вами русификация может отличаться от той, что использовалась при переводе. В этом случае вместо пакета `babel` вам, возможно, понадобится подключать что-то другое. Ожидается, что в будущем `babel` будет стандартным средством локализации \TeX . Проконсультируйтесь с вашим *Local Guide* [4] или с администратором. —**прим. переводчика.**

```
\documentclass{article}
\usepackage[russian]{babel}
\begin{document}
Краткость~--- сестра таланта.
\end{document}
```

Рис. 1.2. Минимальный файл L^AT_EX

```
\documentclass[a4paper,11pt]{article}
\usepackage{latexsym}
\usepackage[russian]{babel}
\begin{document}
\author{Б.~Тоботрас}
\title{Минимализм}
\frenchspacing
\begin{document}
\maketitle
\tableofcontents
\section{Начало}
Вот тут и начинается моя замечательная статья.
\section{Конец}
\ldots{ } а тут она кончается.
\end{document}
```

Рис. 1.3. Пример реалистичной журнальной статьи

1.5. Макеты документов

1.5.1. Классы документов

Первое, что L^AT_EX 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 L^AT_EX 2_ε 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 L^AT_EX может начинаться строкой

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

она заставляет L^AT_EX набирать документ как *статью*, с базовым размером шрифта *одинадцать пунктов* и форматировать документ для *двусторонней* печати на бумаге *формата А4*.

1.5.2. Пакеты

В процессе написания вашего документа, вы, вероятно, обнаружите, что в некоторых областях базовый L^AT_EX не сможет решить ваши проблемы. Если вы захотите включить в документ графику, цветной текст или исходный код программы из внешнего файла, вам нужно будет расширить возможности L^AT_EX. Такие расширения называются пакетами. Пакеты активизируются командой

```
\usepackage[опции]{пакет}
```

Где *пакет* — это имя пакета, а *опции* — список ключевых слов, включающих специ-

Таблица 1.1. Классы документов

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

^aCTAN:macros/latex/packages/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>twocolumn</code>	Заставляет L ^A T _E X набирать документ 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 ne</code> заставляет ваш принтер на самом деле печатать с двух сторон.
<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.

альные свойства пакета. Некоторые пакеты включены в основную поставку L^AT_EX 2_ε (см. таблицу 1.3). Другие предоставляются отдельно. Дополнительная информация об установленных у вас пакетах может содержаться в *Local Guide* [4]. Основной источник информации о пакетах L^AT_EX — это *The L^AT_EX Companion* [3]. Она содержит описания сотен пакетов вместе с информацией о том, как писать ваши собственные расширения для L^AT_EX 2_ε.

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. Описан в <code>syntonly.dtx</code> и в <i>The L^AT_EX Companion</i> [3]. Это удобно для быстрой проверки на ошибки.
<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.

1.5.3. Стили страницы

L^AT_EX 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 L^AT_EX

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 L^AT_EX Companion* [3], а также в разделе 4.4 на странице 53.

1.6. Большие проекты

При работе с большими документами вам может быть удобно разделить входной файл на несколько частей. L^AT_EX 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 L^AT_EX начнет новую страницу, прежде, чем обрабатывать материал, содержащийся в *файле*.

Вторая команда может использоваться только в преамбуле. Она указывает L^AT_EX 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.

Глава 2.

Верстка текста

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

2.1. Структура текста и языка

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.

\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 \LaTeX (и в наборе вообще) — абзац. Он называется «элементом текста», потому что абзац является той типографской формой, в которую облакается 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 \LaTeX , начинают абзацы, не зная этого. Последнюю ошибку особенно легко сделать, когда 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.)

... когда Эйнштейн ввел свою формулу

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

которая является самой широко известной и самой мало понятной физической формулой.

```
\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.
```

... откуда следует закон тока Кирхгоффа:

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

Закон напряжения Кирхгоффа может быть выведен ...

```
\ldots откуда следует
закон тока Кирхгоффа:
\begin{equation}
\sum_{k=1}^n I_k = 0 \ ; \ .
\end{equation}
Закон напряжения Кирхгоффа
может быть выведен \ldots
```

... который имеет несколько преимуществ.

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

является ядром совершенно другой модели транзистора. ...

```
\ldots который имеет
несколько преимуществ.
\begin{equation}
I_D = I_F - I_R
\end{equation}
является ядром совершенно
другой модели транзистора.
\ldots
```

Следующая, более маленькая, единица текста — предложение. В английском тексте после точки, завершающей предложение, ставится бóльший пробел, чем после точки, стоящей после сокращения. ЛАТЭХ 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 эффект записи, например, `\section{The Структура текста и языка}` обычно достаточно очевиден, чтобы было сразу ясно, как использовать эти высокоуровневые структуры.

¹Все это мало применимо для русского, но я решил не выбрасывать этих абзацев из перевода, вдруг кому-то поможет писать на английском, скажем. — прим. переводчика

2.2. Разбиение на строки и страницы

2.2.1. Выровненные абзацы

Книги часто верстаются так, чтобы все строчки имели одинаковую длину. \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 5.3.2.

В некоторых случаях может быть необходимо указать \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 \LaTeX возможность игнорировать вашу команду, если результат будет очень плох. Не путайте эти команды «разрыва (break)» с командами «начала (new)». Даже если вы зададите команду «разрыва», \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.

\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. \LaTeX затем выводит диагностику («overfull hbox») во время обработки входного файла. Чаще всего это случается, когда \LaTeX не может найти место для переноса слова.¹ Давая команду `\sloppy`, вы можете сказать, чтобы \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.

¹Хотя \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.

2.2.2. Переносы

Л^AT_EX переносит слова, когда это необходимо. Если алгоритм переносов не находит правильных точек переноса, вы можете исправить положение, сказав T_EX 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. Регистр букв игнорируется. Следующий пример разрешает переносы в слове «hyphenation», так же, как и в слове «Hyphenation», и запрещает переносы в словах «FORTRAN», «Fortran» и «fortran».

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 Л^AT_EX не переносит такие слова автоматически.

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

Несколько слов можно удержать вместе на одной строке командой

```
\mbox{текст}
```

Она будет в любом случае сохранять свой аргумент вместе.

Номер моего телефона скоро сменится. Он будет 0116 291 2319. Параметр <i>имя файла</i> должен содержать имя файла.	 Номер моего телефона скоро сменится. Он будет\ \mbox{0116 291 2319}. Параметр \mbox{\emph{имя файла}} должен содержать имя файла.
---	--

2.3. Специальные буквы и символы

2.3.1. Знаки кавычек

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

“Пожалуйста, нажмите клавишу ‘х’ ”	‘ ‘Пожалуйста, нажмите клавишу ‘х’ ’ ’
------------------------------------	--

2.3.2. Тире и дефисы

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

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

2.3.3. 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, называемаяся

\ldots

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

2.3.4. Лигатуры

Некоторые комбинации букв верстаются не просто набором разных букв друг за другом, а с использованием специальных символов¹.

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

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

He “shelfful” а “shelfful”	He ‘ ‘shelfful’ ’\\ а ‘ ‘shelf\mbox{}ful’ ’
-------------------------------	--

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

2.3.5. Акценты и специальные символы

L^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.

Hôtel, naïve, élève,		H\^otel, na\"i ve, \'el\'eve,\\
smørrebrød, ¡Señorita!,		sm\o rrebr\o d, !'Se\~norita!,\\
Schönbrunner Schloß Straße		Sch\"onbrunner Schlo\ss{}
		Stra\ss e

Таблица 2.1. Акценты и специальные символы

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

2.4. Поддержка иностранных языков

Если вам нужно писать документы на отличных от английского языках, то L^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) L^AT_EX должен знать правила переноса для нового языка. Подключение правил переноса к L^AT_EX более сложно. Оно включает перестроение форматного файла с другими разрешенными образцами переноса. Ваш *Local Guide* [4] должен содержать больше информации об этом.

¹Содержание, Список иллюстраций, Библиография ...

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* [4]. `Babel` автоматически активизирует правильные правила переноса для выбираемых вами языков. Если форматный файл вашего `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.

Некоторые компьютерные системы позволяют вам вводить специальные символы прямо с клавиатуры. `LaTeX` может обрабатывать такие символы. Начиная с выпуска `LaTeX 2ε` от декабря 1994г., поддержка нескольких входных кодировок является частью базовой поставки `LaTeX 2ε`. Ознакомьтесь с пакетом `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 на IBM PC кодируется как 132, а на некоторых Unix системах, использующих ISO-LATIN 1, — как 228. Поэтому используйте эту возможность с осторожностью.

Иной случай — с кодировкой шрифта. Она определяет, в какой позиции в пределах шрифта `TeX` находится каждая буква. Оригинальный `TeX` шрифт `Computer Modern` содержит только 128 символов из старого 7-битного набора символов ASCII. Когда требуются акцентированные символы, `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.

К счастью, большинство современных поставок `TeX` содержат копию шрифтов EC. Эти шрифты выглядят, как шрифты `Computer Modern`, но содержат специальные символы для большинства акцентированных букв, используемых в европейских языках. Используя эти шрифты, вы можете улучшить переносы в не-английских документах. Шрифты EC активизируются включением в преамбуле вашего документа пакета `fontenc`.

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

2.5. Пробелы между словами

Для получения ровного правого края вывода `LaTeX` вставляет различные интервалы между словами. В конце предложения он вставляет слегка больший интервал, делая текст более читабельным¹. `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.

¹В соответствии с традициями верстки, принятыми в английском языке — прим. переводчика.

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

Дополнительный пробел после точек можно запретить командой

```
\frenchspacing
```

которая указывает L^AT_EX 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.

2.6. Заголовки, главы и разделы

Чтобы помочь читателю ориентироваться в вашей работе, вы должны разделять ее на главы, разделы и подразделы. L^AT_EX поддерживает это специальными командами, принимающими в качестве аргумента заголовки раздела. Ваше дело — использовать их в надлежащем порядке.

Класс `article` включает следующие команды секционирования:

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

В классах `report` и `book` вы можете использовать две дополнительные команды:

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

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 L^AT_EX 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 цифр¹.

¹В классе `article` меняется нумерация разделов.

L^AT_EX создает оглавление, беря заголовки разделов и номера страниц из предыдущего прохода по документу. Команда

```
\tableofcontents
```

вставляет оглавление в то место, где она вызвана. Чтобы получить правильное оглавление, новый документ должен быть обработан L^AT_EX дважды. В особых случаях может быть необходим и третий проход. Когда это потребуется, L^AT_EX 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[Прочтите! Это замечательно]{Это~--- длинный,  
длинный и очень нудный заголовок}
```

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 1.3 на странице 7.

Помимо описанных выше команд секционирования, L^AT_EX 2_ε вводит три дополнительных команды для использования с классом `book`.

```
\frontmatter, \mainmatter и \backmatter
```

Они полезны для деления вашей публикации. Команды изменяют заголовки глав и нумерацию страниц так, как это ожидается от книги.¹

2.7. Перекрестные ссылки

В книгах, отчетах и статьях часто встречаются перекрестные ссылки на иллюстрации, таблицы и отдельные части текста. Для этого L^AT_EX предоставляет следующие команды:

¹`\frontmatter` начинает вводную часть книги, `\mainmatter` — основную, `\backmatter` — заключающую. — прим. переводчика

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

где *метка* — выбранный пользователем идентификатор. L^AT_EX заменяет `\ref` номером раздела, подраздела, иллюстрации, таблицы или уравнения, где была использована соответствующая команда `\label`. `\pageref` печатает номер страницы, соответствующей команде `\label`¹. Так же, как и в случае с заголовками разделов, здесь также используются номера из предыдущего прохода.

Ссылка на этот раздел выглядит так:
«см. раздел 2.7 на стр. 22.»

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

2.8. Сноски

Команда

```
\footnote{текст сноски}
```

печатает сноску внизу текущей страницы. Сноски всегда должны помещаться после слова, к которому они относятся.

Пользователи L^AT_EX часто употребляют сноски².

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

2.9. Выделенные слова

В рукописи, напечатанной на машинке, **важные слова выделяются подчеркиванием**. В печатных изданиях 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.

²Это — сноска.

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

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

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

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

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

2.10. 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.

2.10.1. Список, перечисление и описание

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

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.

```
\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}
```

2.10.2. 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, `ЛATEX` определит их автоматически.

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

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

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

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

В центре
Земли

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

2.10.3. Цитаты и стихи

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

Типографское правило для длины строки:

Никакая строка не должна содержать больше 66 символов.

Вот почему \LaTeX делает такими широкими поля страниц.

Поэтому в газетах часто применяют набор в несколько колонок.

Типографское правило для длины строки:

`\begin{quote}`
Никакая строка не должна содержать больше 66~символов.

Вот почему `\LaTeX{}` делает такими широкими поля страниц.
`\end{quote}`

Поэтому в газетах часто применяют набор 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:

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.

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}`

2.10.4. 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 может использовать любой символ, кроме букв, «*» или пробела. Многие примеры на \LaTeX в этом буклете набраны этой командой.

Команда `\ldots` ...

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

вариант окружения
`verbatim` со
звездочкой выделяет
пробелы в тексте

Команда `\verb|\ldots| \ldots`

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

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

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

```
во\verb*|так :-)\
```

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

2.10.5. Таблицы

Окружение `tabular` используют для верстки таблиц, возможно, с горизонтальными и вертикальными линиями. L^AT_EX автоматически определяет ширину столбцов.

Аргумент *спецификация* команды

```
\begin{tabular}{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.

Внутри окружения `tabular` знак «&» переходит к следующему столбцу, команда `\\` начинает новую строку, а `\hline` вставляет горизонтальную линию.

54	шестнадцатеричное
124	восьмеричное
1010100	двоичное
84	десятичное

```
\begin{tabular}{|r|l|}
\hline
54 & шестнадцатеричное \\
124 & восьмеричное \\
1010100 & двоичное \\
\hline \hline
84 & десятичное \\
\hline
\end{tabular}
```

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

```

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

```

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 `@{}`:

нет ведущего пробела

```

\begin{tabular}{@{} 1 @{}}
\hline
нет ведущего пробела\\
\hline
\end{tabular}

```

ведущий пробел слева и справа

```

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

```

Поскольку встроенный способ выровнять числовые столбцы по десятичной точке отсутствует¹, 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`:

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

```

\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}

```

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

2.11. 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 уместяющаяся на текущей странице, может ‘плавать’, перемещаясь на следующую страницу в процессе заполнения текстом текущей. L^AT_EX предлагает для плавающих объектов два окружения, одно для таблиц и одно для иллюстраций. Чтобы полностью использовать их преимущества, важно примерно представлять, как L^AT_EX обрабатывает плавающие объекты. Иначе они могут стать источником разочарования из-за того, что L^AT_EX помещает их не туда, куда вы хотите.

Давайте вначале рассмотрим команды, предоставляемые L^AT_EX 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 L^AT_EX, куда можно перемещать плавающий объект. *Спецификация размещения* конструируется путем собирания в строчку *ключей размещения плавающего объекта*. См. таблицу 2.2.

Например, таблицу можно начать следующей строкой:

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

Спецификация размещения `[!hbp]` позволяет L^AT_EX разместить таблицу прямо по месту (**h**), или внизу той же страницы (**b**), или на выделенной странице (**p**), и все это —

Таблица 2.2. 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

даже если это будет смотреться не так уж хорошо (!). Если никакой спецификации размещения не задано, стандартные классы предполагают `[tbp]`.

\LaTeX 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 таблиц¹. Когда начинается новая страница, \LaTeX проверяет, можно ли заполнить специальную страницу плавающими объектами из очередей. Если нет, то первый объект из каждой очереди считается только что встретившимся в тексте: \LaTeX 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. \LaTeX сохраняет порядок, в котором встретились плавающие объекты соответствующего типа. Поэтому иллюстрация, которую не удастся разместить, отталкивает все дальнейшие иллюстрации к концу документа. Следовательно:

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

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 \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]{Ддддллллллиииннннныыыыыыыыыййй}
```

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.

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

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

```
\end{figure}  
\
```

В этом примере L^AT_EX 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, L^AT_EX начинает новую страницу и снова рассматривает иллюстрацию, как если бы она только что появилась в тексте.

В определенных случаях может быть необходимо использовать команду

```
\clearpage или даже \cleardoublepage
```

Она указывает L^AT_EX немедленно разместить все плавающие объекты, оставшиеся в очередях, и затем начать новую страницу. `\cleardoublepage`, помимо этого, начинает новую левостороннюю страницу.

Позже вы узнаете, как включать в ваши документы L^AT_EX рисунки в формате POSTSCRIPT.

¹Предполагая, что очередь иллюстраций пуста

Глава 3.

Набор математических формул

Вот теперь мы готовы! В этой главе мы встретимся с основной мощью \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 AMS- \LaTeX ¹ или в каком-нибудь ином пакете.

3.1. Общие сведения

\LaTeX включает в себя специальный режим для верстки математики. Математический текст внутри абзаца вводится между $\backslash($ и $\backslash)$, между $\$$ и $\$$ или между $\backslashbegin{\math}$ или $\backslashend{\math}$.

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

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

\TeX произносится как $\tau\epsilon\chi$.

100 м³ воды.

Это исходит от моего ♡.

$\backslashTeX\{\}$ произносится как $\backslash\tau\backslash\epsilon\backslash\chi\backslash\{\}$.
100~м³ воды.
Это исходит от моего $\backslash\heartsuit\{\}$.

Большие математические уравнения или формулы предпочтительнее «выключать», то есть верстать их на отдельных строках. Для этого заключайте их между $\backslash[$ и $\backslash]$ или между $\backslashbegin{\displaymath}$ и $\backslashend{\displaymath}$. Это дает вам не нумерованные формулы. Если вы хотите, чтобы \LaTeX их нумеровал, пользуйтесь окружением `equation`.

¹CTAN:macros/latex/packages/amslatex

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

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

И еще одна строчка

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}
```

И еще одна строчка

При помощи `\label` и `\ref` вы можете в тексте сослаться на уравнение.

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

```
\begin{equation}
\label{eq:eps}
\epsilon > 0
\end{equation}
Из (\ref{eq:eps})
выводим \ldots
```

Заметьте, что выражения в выключных формулах верстаются в другом стиле:

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

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

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

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

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 `\text{rm}\{...\}`.

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

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

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

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

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:

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

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

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: `{...}`.

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

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

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 3.9 на странице 42.

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

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

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

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

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

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

Квадратный корень вводится как `\sqrt`, корень n -ной степени печатается при помощи `\sqrt[n]`. Размер знака корня выбирается ЛАТЭХ автоматически. Если нужен

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

один только знак, используйте `\surd`.

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

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.

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

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.

$y = x^2$	$y' = 2x$	$y'' = 2$	$\backslash \text{begin}\{\text{displaymath}\}$ $y=x^2 \backslash \text{quad} y'=2x$ $\backslash \text{quad} y''=2$ $\backslash \text{end}\{\text{displaymath}\}$
-----------	-----------	-----------	--

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`.

\vec{a}	\overrightarrow{AB}	$\backslash \text{begin}\{\text{displaymath}\}$ $\vec{a} \backslash \text{quad}$ \overrightarrow{AB} $\backslash \text{end}\{\text{displaymath}\}$
-----------	-----------------------	---

Имена функций типа `\lg` часто верстаются прямым шрифтом, а не курсивом, как переменные. Поэтому L^AT_EX содержит следующие команды для набора имен наиболее важных функций:

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

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

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

Двухъярусная **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’.

<pre>\$1\frac{1}{2}\$~часа \begin{displaymath} \frac{x^2}{k+1}\qquad x^{\frac{2}{k+1}}\qquad x^{1/2} \end{displaymath} \newpage\begin{verbatim}</pre>	<pre>\$1\frac{1}{2}\$~часа \begin{displaymath} \frac{x^2}{k+1}\qquad x^{\frac{2}{k+1}}\qquad x^{1/2} \end{displaymath}</pre>
---	--

Для верстки биномиальных коэффициентов или аналогичных структур можно пользоваться или `\dots \choose \dots`, или `\dots \atop \dots`. Вторая команда дает тот же вывод, что и первая, но без скобок.

$\binom{n}{k} \quad x$ $y + 2$	<pre>\begin{displaymath} {n \choose k}\qquad {x \atop y+2} \end{displaymath}</pre>
--------------------------------	--

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

$\sum_{i=1}^n \int_0^{\frac{\pi}{2}}$	<pre>\begin{displaymath} \sum_{i=1}^n \qquad \int_0^{\frac{\pi}{2}} \end{displaymath}</pre>
---------------------------------------	---

Для **скобок** и прочих ограничителей в \TeX существует множество символов (скажем, $[\langle \| \Downarrow]$). 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 44.

<pre>\begin{displaymath} \{a,b,c\}\neq\{a,b,c\} \end{displaymath} \newpage\begin{verbatim}</pre>	<pre>\begin{displaymath} \{a,b,c\}\neq\{a,b,c\} \end{displaymath}</pre>
--	---

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

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

<pre>\newcommand{\ud}{\mathrm{d}} \begin{displaymath} \iint_{D} \, \, \, \ud x \, \, \, \ud y \end{displaymath} \newpage\begin{verbatim}</pre>	<pre>\newcommand{\ud}{\mathrm{d}} \begin{displaymath} \iint_{D} \, \, \, \ud x \, \, \, \ud y \end{displaymath}</pre>
--	---

Детали смотрите в электронном документе `textmath.tex` (распространяется с $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$) или в главе 8 “The $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ Companion”¹.

3.5. Вертикально расположенный материал

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

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

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

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

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

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

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 `{rcl}`, где средний столбец используется для знака равенства, или знака неравенства, или другого подходящего знака. Команда `\\` разбивает строки.

¹Можно получить на `CTAN:info/ch8.*`

		<pre>\begin{eqnarray}</pre>
$f(x) = \cos x$	(3.5)	<pre>f(x) & = & \cos x & \quad \backslash\backslash</pre>
$f'(x) = -\sin x$	(3.6)	<pre>f'(x) & = & -\sin x & \quad \backslash\backslash</pre>
$\int_0^x f(y)dy = \sin x$	(3.7)	<pre>\int_{0}^{x} f(y)dy & \quad & \backslash\backslash</pre> <pre>= & \sin x</pre> <pre>\end{eqnarray}</pre>

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:

		<pre>{\setlength\arraycolsep{2pt}}</pre> <pre>\begin{eqnarray}</pre>
$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} -$	(3.8)	<pre>\sin x & = & x -\frac{x^{3}}{3!}</pre> <pre>+\frac{x^{5}}{5!}-\{}</pre> <pre>\nonumber\backslash</pre> <pre>& & \{-\frac{x^{7}}{7!}+\}\cdots</pre> <pre>\end{eqnarray}}</pre>
$-\frac{x^7}{7!} + \dots$		
		<pre>\begin{eqnarray}</pre> <pre>\lefteqn{\cos x = 1</pre> <pre>-\frac{x^{2}}{2!} +\{}</pre> <pre>\nonumber\backslash</pre> <pre>& & \{+\frac{x^{4}}{4!}</pre> <pre>-\frac{x^{6}}{6!}+\}\cdots</pre> <pre>\end{eqnarray}</pre>
$\cos x = 1 - \frac{x^2}{2!} +$	(3.9)	
$+\frac{x^4}{4!} - \frac{x^6}{6!} + \dots$		

Команда `\nonumber` заставляет \LaTeX не генерировать номер для этого уравнения.

Такими методами может быть сложно получить правильно выглядящие вертикально выровненные уравнения; более мощную альтернативу предоставляет пакет `amsmath`.

3.6. Размер математического шрифта

В математическом режиме \TeX выбирает размер шрифта в зависимости от контекста. Индексы, например, верстаются меньшим шрифтом. Если вы хотите добавить к уравнению обычный текст, не пользуйтесь командой `\textrm`, так как механизм переключения размера работать не будет, потому что `\textrm` временно выходит в текстовый режим. Чтобы оставить его работающим, используйте команду `\mathrm`. Но имейте в виду, `\mathrm` будет хорошо работать только с короткими элементами. Пробелы по-прежнему не активны и акцентированные символы не работают¹.

¹При подключении пакета \mathcal{AMS} - \TeX команда `\textrm` начинает работать с изменением размера.

$$2^{\text{nd}} \quad 2^{\text{nd}} \quad (3.10) \quad \left| \begin{array}{l} \backslash\text{begin}\{\text{equation}\} \\ 2^{\text{\texttrm{nd}}} \quad \backslash\text{quad} \\ 2^{\text{\mathrm{nd}}} \\ \backslash\text{end}\{\text{equation}\} \end{array} \right.$$

Тем не менее, иногда вам может быть нужно указать L^AT_EX точный размер шрифта. В математическом режиме размер устанавливается четырьмя командами:

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

Смена стилей влияет также на способ изображения пределов.

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

$$\left| \begin{array}{l} \backslash\text{begin}\{\text{displaymath}\} \\ \text{\mathop{\mathrm{corr}}}(X, Y) = \\ \backslash\text{frac}\{\text{\displaystyle} \\ \quad \sum_{i=1}^n (x_i - \overline{x}) \\ \quad (y_i - \overline{y})\} \\ \quad \{\text{\displaystyle}\biggl[\\ \quad \sum_{i=1}^n (x_i - \overline{x})^2 \\ \quad \sum_{i=1}^n (y_i - \overline{y})^2 \\ \quad \biggr]^{\text{\scriptsize 1/2}}\} \\ \backslash\text{end}\{\text{displaymath}\} \end{array} \right.$$

Это один из примеров, когда нам нужны скобки большие, чем предоставляемые стандартными `\left[\quad \right]`.

3.7. Теоремы, законы, ...

При написании математических документов, вам, вероятно, нужен способ верстки «лемм», «определений», «аксиом» и аналогичных структур. L^AT_EX поддерживает это командами

```
\newtheorem{название}[счетчик]{текст}[раздел]
```

Аргумент *название* — это краткое ключевое слово, используемое для идентификации «теоремы». Аргументом *текст* вы определяете настоящее название «теоремы», под которым она будет печататься в документе.

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:

Law 1. <i>Don't hide in the witness box</i>	<pre>% определения для % преамбулы документа \newtheorem{law}{Law} \newtheorem{jury}[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}</pre>
Jury 2 (The Twelve). <i>It could be you!</i> <i>So beware and see law 1</i>	
Law 3. <i>No, No, No</i>	

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.

Murphy 3.7.1. <i>Если существует два или более способа сделать нечто, и один из этих способов может привести к катастрофе, то кто-то обязательно это сделает.</i>	<pre>\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}</pre>
--	--

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

3.8. Полу жирные символы

В \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.

μ, M	\mathbf{M}	$\boldsymbol{\mu}, \boldsymbol{M}$	<pre>\begin{displaymath} \mu, M \quad \mathbf{M} \quad \boldsymbol{\mu}, \boldsymbol{M} \end{displaymath}</pre>
----------	--------------	------------------------------------	---

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

Пакет `amsbsy` (включаемый пакетом `amsmath`) включает команду `\boldsymbol`.

μ, M **μ, M**

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

3.9. Список математических символов

В следующих таблицах вы найдёте 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:macros/latex/packages/amslatex`

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

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 ¹	<code>\sqsubset</code> ¹	\sqsupset ¹	<code>\sqsupset</code> ¹	\bowtie	<code>\bowtie</code> ¹
\sqsubseteq	<code>\sqsubseteq</code>	\sqsupseteq	<code>\sqsupseteq</code>	\bowtie	<code>\bowtie</code>
\in	<code>\in</code>	\ni , \owns	<code>\ni</code> , <code>\owns</code>	\propto	<code>\propto</code>
\vdash	<code>\vdash</code>	\dashv	<code>\dashv</code>	\models	<code>\models</code>
$ $	<code>\mid</code>	\parallel	<code>\parallel</code>	\perp	<code>\perp</code>
\smile	<code>\smile</code>	\frown	<code>\frown</code>	\asymp	<code>\asymp</code>
$:$	<code>:</code>	\notin	<code>\notin</code>	\neq или \ne	<code>\neq</code> или <code>\ne</code>

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

Таблица 3.5. Бинарные операторы

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

Таблица 3.6. Большие операторы

\sum	<code>\sum</code>	\cup	<code>\bigcup</code>	\vee	<code>\bigvee</code>	\oplus	<code>\bigoplus</code>
\prod	<code>\prod</code>	\cap	<code>\bigcap</code>	\wedge	<code>\bigwedge</code>	\otimes	<code>\bigotimes</code>
\coprod	<code>\coprod</code>	\sqcup	<code>\bigsqcup</code>			\odot	<code>\bigodot</code>
\int	<code>\int</code>	\oint	<code>\oint</code>			\oplus	<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>	\Lleftarrow	<code>\Lleftarrow</code>	\Uparrow	<code>\Uparrow</code>
\Rightarrow	<code>\Rightarrow</code>	\Rrightarrow	<code>\Rrightarrow</code>	\Downarrow	<code>\Downarrow</code>
\Leftrightarrow	<code>\Leftrightarrow</code>	\Leftrightarrow	<code>\Leftrightarrow</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	<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>] или \rbrack</code>	\downarrow	<code>\downarrow</code>	\Downarrow	<code>\Downarrow</code>
$\{$	<code>\{</code> или <code>\lbrace</code>	$\}$	<code>\} или \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>\lgroup</code>	$\right)$	<code>\rgroup</code>	$\left\{$	<code>\lmoustache</code>	$\right\}$	<code>\rmoustache</code>
$\left\{$	<code>\arrowvert</code>	$\ $	<code>\Arrowvert</code>	$ $	<code>\bracevert</code>		

Таблица 3.10. Прочие символы

\dots	<code>\dots</code>	\cdots	<code>\cdots</code>	\vdots	<code>\vdots</code>	\ddots	<code>\ddots</code>
\hbar	<code>\hbar</code>	\imath	<code>\imath</code>	\jmath	<code>\jmath</code>	ℓ	<code>\ell</code>
\Re	<code>\Re</code>	\Im	<code>\Im</code>	\aleph	<code>\aleph</code>	\wp	<code>\wp</code>
\forall	<code>\forall</code>	\exists	<code>\exists</code>	\mho 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>	\square	<code>\Box</code> ¹	\diamond	<code>\Diamond</code> ¹
\bot	<code>\bot</code>	\top	<code>\top</code>	\angle	<code>\angle</code>	\surd	<code>\surd</code>
\diamond	<code>\diamondsuit</code>	\heartsuit	<code>\heartsuit</code>	\clubsuit	<code>\clubsuit</code>	\spadesuit	<code>\spadesuit</code>
\neg	<code>\neg</code> 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.

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

Таблица 3.12. Ограничители AMS

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

Таблица 3.13. Буквы греческого и иврита AMS

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

Таблица 3.14. Бинарные отношения AMS

\lessdot	<code>\lessdot</code>	\gtrdot	<code>\gtrdot</code>	\doteqdot или <code>\Doteq</code>	<code>\doteqdot</code> или <code>\Doteq</code>
\leqslant	<code>\leqslant</code>	\geqslant	<code>\geqslant</code>	\risingdotseq	<code>\risingdotseq</code>
\leqslantless	<code>\leqslantless</code>	\geqslantgtr	<code>\geqslantgtr</code>	\fallingdotseq	<code>\fallingdotseq</code>
\leqq	<code>\leqq</code>	\geqq	<code>\geqq</code>	\eqcirc	<code>\eqcirc</code>
\lll 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>	\Vvdash	<code>\Vvdash</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>	\upuparrows	<code>\upuparrows</code>
\leftrightarrows	<code>\leftrightarrows</code>	\rightleftarrows	<code>\rightleftarrows</code>	\downdownarrows	<code>\downdownarrows</code>
\Lleftarrow	<code>\Lleftarrow</code>	\Rrightarrow	<code>\Rrightarrow</code>	\upharpoonleft	<code>\upharpoonleft</code>
\twoheadleftarrow	<code>\twoheadleftarrow</code>	\twoheadrightarrow	<code>\twoheadrightarrow</code>	\upharpoonright	<code>\upharpoonright</code>
\leftarrowtail	<code>\leftarrowtail</code>	\rightarrowtail	<code>\rightarrowtail</code>	\downharpoonleft	<code>\downharpoonleft</code>
\leftrightharpoons	<code>\leftrightharpoons</code>	\rightleftharpoons	<code>\rightleftharpoons</code>	\downharpoonright	<code>\downharpoonright</code>
\Lsh	<code>\Lsh</code>	\Rsh	<code>\Rsh</code>	\rightsquigarrow	<code>\rightsquigarrow</code>
\looparrowleft	<code>\looparrowleft</code>	\looparrowright	<code>\looparrowright</code>	\leftrightsquigarrow	<code>\leftrightsquigarrow</code>
\curvearrowleft	<code>\curvearrowleft</code>	\curvearrowright	<code>\curvearrowright</code>		
\circlearrowleft	<code>\circlearrowleft</code>	\circlearrowright	<code>\circlearrowright</code>		

Таблица 3.16. Отрицательные бинарные отношения и стрелки AMS

\nless	\ngtr	\varsubsetneqq
\lneq	\gneq	\varsupsetneqq
\nleq	\ngeq	\subsetneqq
\nleqslant	\ngeqslant	\supsetneqq
\lneqq	\gneqq	\mid
\lvertneqq	\gvertneqq	\parallel
\nleqq	\ngeqq	\shortmid
\lnsim	\gnsim	\shortparallel
\lnapprox	\gnapprox	\sim
\nprec	\nsucc	\cong
\npreceq	\nsucceq	\nvdash
\precneqq	\succneqq	\nVDash
\precnsim	\succnsim	\nVdash
\precnapprox	\succnapprox	\nVDash
\subsetneq	\supsetneq	\ntriangleleft
\varsubsetneq	\varsupsetneq	\ntriangleright
\subsetseteq	\nsupseteq	\ntrianglelefteq
\subsetneqq	\supsetneqq	\ntrianglerighteq
\nleftarrow	\nrightarrow	\leftrightharrow
\nLeftarrow	\nRightarrow	\Leftrightarrow

Таблица 3.17. Бинарные операторы AMS

$\dot{+}$	\cdot	\intercal
\ltimes	\rtimes	\div
\Cup или \doublecup	\Cap или \doublecap	\smallsetminus
\veebar	$\bar{\wedge}$	$\bar{\bar{\wedge}}$
\boxplus	\boxminus	\circleddash
\boxtimes	\boxdot	\circledcirc
\leftthreetimes	\rightthreetimes	\circledast
\curlyvee	\curlywedge	

Таблица 3.18. Прочие символы AMS

\hbar	<code>\hbar</code>	\hbar	<code>\hslash</code>	\mathbb{k}	<code>\Bbbk</code>
\square	<code>\square</code>	\blacksquare	<code>\blacksquare</code>	\textcircled{S}	<code>\circledS</code>
\triangle	<code>\vartriangle</code>	\blacktriangle	<code>\blacktriangle</code>	\complement	<code>\complement</code>
∇	<code>\triangledown</code>	\blacktriangledown	<code>\blacktriangledown</code>	\Game	<code>\Game</code>
\lozenge	<code>\lozenge</code>	\blacklozenge	<code>\blacklozenge</code>	\bigstar	<code>\bigstar</code>
\sphericalangle	<code>\angle</code>	\sphericalangle	<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>	
$ABCdef$	<code>\mathnormal{ABCdef}</code>	
ABC	<code>\mathcal{ABC}</code>	
	<code>\mathcal{ABC}</code>	eucal с опцией: <code>mathcal</code> или
	<code>\mathscr{ABC}</code>	eucal с опцией: <code>mathscr</code>
$\frac{ABC}{def}$	<code>\mathfrac{ABCdef}</code>	eufrak
\mathbb{ABC}	<code>\mathbb{ABC}</code>	amfonts или amssymb

Глава 4.

Специальные возможности

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

4.1. Включение EPS графики

\LaTeX имеет базовые средства для работы с плавающими объектами, такими, как иллюстрациями и таблицами, при помощи окружений `figure` и `table`.

Существует также несколько способов создавать собственно графику средствами базового \LaTeX или его расширений. К сожалению, большинство пользователей находит их трудными для понимания. Дополнительная информация приведена в *The \LaTeX Companion* [3] и в *\LaTeX Manual* [1].

Более простой метод получения графики в документе — это создавать ее специализированными программными пакетами¹ и включать в документ готовую графику. Пакеты \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 принтер².

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:support/ghostscript`.

³`CTAN:macros/latex/packages/graphics`.

⁴Если ваша программа не имеет экспорта в EPS, можно попробовать установить драйвер PostScript принтера (скажем, какого-нибудь Apple LaserWriter) и печатать в файл через этот драйвер. Если вам

- 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[ключ=значение, ... ]{файл}
```

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

Таблица 4.1. Названия ключей для пакета `graphicx`

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

Возможно, пример сделает это понятнее:

```
\begin{figure}
\begin{center}
\includegraphics[angle=90, width=0.5\textwidth]{test.eps}
\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 высоты. Параметры высоты и ширины могут также быть указаны в абсолютных размерах. Обратитесь к таблице 5.5 на странице 62. Если вы хотите знать об этом больше, прочтите [8] и [11].

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.

4.2. Библиография

Окружение `thebibliography` генерирует библиографию. Каждый элемент начинается с

```
\bibitem{маркер}
```

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

```
\cite{маркер}
```

Элементы библиографии нумеруются автоматически. Параметр после команды `\begin{thebibliography}` устанавливает максимальную ширину этих номеров. В следующем примере `{99}` указывает L^AT_EX, что ни один из номеров элементов не будет шире, чем число ‘99’.

Partl [1] предложил, чтобы ...

Литература

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

Partl~\cite{pa} предложил,
чтобы \ldots

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

Для использования в больших проектах обратите внимание на программу BibT_EX. Она входит в большинство поставок T_EX. BibT_EX 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 BibT_EX, 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>	<i>Sam</i> , 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

4.3. Указатели

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

Для включения возможностей \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 \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 \LaTeX , но другое расширение имени (`.idx`). Если теперь повторно обработать входной файл, этот отсортированный указатель включается в документ в то место, где \LaTeX находит команду

```
\printindex
```

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

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

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.

```

\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
\fancyhead[plain]{%
  \fancyhead{} % на обычных страницах убираем колонтитулы
  \renewcommand{\headrulewidth}{0pt} % и линейку
}

```

Рис. 4.1. Пример настройки `fancyhdr`

Сложность в настройке колонтитулов в том, чтобы включить туда вещи вроде заголовков раздела или главы. \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, называющуюся `\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 ссылке.

¹Доступный из `CTAN:macros/latex/contrib/supported/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 замечательно¹, 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 [9].

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. переводчика

Глава 5.

Настройка L^AT_EX

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 L^AT_EX 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 L^AT_EX новым вещам, и тому, как сделать его вывод отличающимся от того, что производится по умолчанию.

5.1. 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 L^AT_EX, автор создал пакет, в котором определил новые команды и окружения для этой цели. Теперь можно просто писать:

`\dum`

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

В этом примере используются как новое окружение, называющееся `command` и отвечающее за рисование рамки вокруг команды, так и новая команда, называющаяся `\ci` и верстающая название команды и заносщая 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, сверстанные в рамке, он просто изменит определение окружения `command`. Это намного проще, чем пройти по всему документу, выискивая все места, где использованы общие команды L^AT_EX для рисования рамки вокруг слов.

5.1.1. 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 L^AT_EX 2_ε”. Такая команда пригодится, если вам много раз приходится писать название этой книги.

“The not so Short Introduction to L ^A T _E X 2 _ε ” ... “The not so Short Introduction to L ^A T _E X 2 _ε ”	<pre>\newcommand{\tnss}{The not so Short Introduction to \LaTeXe} % в теле документа: ‘‘\tnss’’ \ldots{ } ‘‘\tnss’’</pre>
---	---

Следующий пример иллюстрирует использование аргумента *число*. Метка `#1` заменяется на заданный аргумент. Если вы хотите использовать более одного аргумента, пользуйтесь `#2`, и так далее.

<ul style="list-style-type: none"> • The <i>He</i> очень краткое введение в L^AT_EX 2_ε • The <i>Очень</i> краткое введение в L^AT_EX 2_ε 	<pre>\newcommand{\txsit}[1] {The \emph{#1} краткое введение в \LaTeXe} % в теле документа \begin{itemize} \item \txsit{He очень} \item \txsit{Очень} \end{itemize}</pre>
---	--

L^AT_EX 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 L^AT_EX 2_ε ее молча проигнорирует.

5.1.2. Новые окружения

Аналогично команде `\newcommand`, существует команда для создания вашего собственного окружения. Команда `\newenvironment` имеет следующий синтаксис:

```
\newenvironment{название}[номер]{начало}{конец}
```

Подобно команде `\newcommand`, `\newenvironment` можно использовать с необязатель-

ным 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`.

■	Мои смиренные подданные ...	■	<pre> \newenvironment{king} {\rule{1ex}{1ex}% \hspace{\stretch{1}}} {\hspace{\stretch{1}}% \rule{1ex}{1ex}} \begin{king} Мои смиренные подданные\ldots \end{king} </pre>
---	-----------------------------	---	---

Аргумент *номер* используют так же, как и для команды `\newcommand`. ЛАТ_ЭX 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. 67, команда `\stretch` описана на стр. 62, а описание команды `\hspace` находится на стр. 61.

5.1.3. Ваш собственный пакет

Когда вы определяете множество новых окружений и команд, преамбулы ваших документов становятся очень большими. В этой ситуации представляется разумным создать пакет ЛАТ_ЭX, содержащий определения всех ваших команд и окружений. Потом можно командой `\usepackage` использовать пакет в ваших документах.

```

% 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. 5.1. Пример пакета

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 ЛАТ_ЭX название пакета, что позволяет ему выдавать осмысленное сообщение об ошибке, когда вы пыта-

есть включать пакет дважды. Иллюстрация 5.1 показывает маленький пример пакета, содержащего определенные в вышеприведенных примерах команды.

5.2. Шрифты и их размеры

5.2.1. Команды смены шрифта

L^AT_EX 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 5.1 и 5.2. Действительный размер каждого шрифта определяется дизайном и зависит от класса и опций документа. Таблица 5.3 показывает абсолютные размеры, соответствующие этим командам в стандартных классах документов.

The small and bold Romans ruled all of great big <i>Italy</i> .	<pre>{\small The small and \textbf{bold} Romans ruled} {\Large all of great big \textit{Italy}.}</pre>
--	--

Важная особенность L^AT_EX_{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 5.4.

В связи с командами смены размера шрифта заметную роль играют фигурные скобки. Они используются для построения *групп*. Группы ограничивают область действия большинства команд L^AT_EX.

Ему нравятся БОЛЬШИЕ И маленькие БУКВЫ.	<pre>Ему нравятся {\LARGE большие и {\small маленькие} буквы}.</pre>
---	--

Команды, влияющие на размер шрифта, влияют также на расстояние между стро-

Таблица 5.1. Шрифты

<code>\textrm{...}</code>	прямой	<code>\textsf{...}</code>	без засечек
<code>\texttt{...}</code>	пишущая машинка		
<code>\textmd{...}</code>	нормальный	<code>\textbf{...}</code>	полужирный
<code>\textup{...}</code>	прямой	<code>\textit{...}</code>	<i>курсив</i>
<code>\textsl{...}</code>	<i>наклонный</i>	<code>\textsc{...}</code>	КАПИТЕЛЬ
<code>\emph{...}</code>	<i>выделенный</i>	<code>\textnormal{...}</code>	обычный

Таблица 5.2. 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 5.3. Абсолютные размеры шрифтов в стандартных классах

<i>Размер</i>	<i>10pt (по умолчанию)</i>	<i>опция 11pt</i>	<i>опция 12pt</i>
<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

Таблица 5.4. Математические шрифты

<i>Команда</i>	<i>Пример</i>	<i>Вывод</i>
<code>\mathcal{...}</code>	<code>\$\$\mathcal{B}=c\$</code>	$\mathcal{B} = c$
<code>\mathrm{...}</code>	<code>\$\$\mathrm{K}_2\$</code>	K_2
<code>\mathbf{...}</code>	<code>\$\$\sum x=\mathbf{v}\$</code>	$\sum x = \mathbf{v}$
<code>\mathsf{...}</code>	<code>\$\$\mathsf{G\times R}\$</code>	$G \times R$
<code>\mathtt{...}</code>	<code>\$\$\mathtt{L}(b,c)\$</code>	$L(b, c)$
<code>\mathnormal{...}</code>	<code>\$\$\mathnormal{R_{19}}\neq R_{19}\$</code>	$R_{19} \neq R_{19}$
<code>\mathit{...}</code>	<code>\$\$\mathit{ffi}\neq ffi\$</code>	$ffi \neq ffi$

ками, 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 примерах:

Не читайте это! Это неправ- да. Верьте мне!	{\Large Не читайте это! Это неправда. Верьте мне!}\par}
--	--

Это тоже неправда. Но по- мните, что я вру.	{\Large Это тоже неправда. Но помните, что я вру.}\par}
--	--

Если вы хотите применить команду изменения размера к целому абзацу текста или больше того, то для этого лучше использовать синтаксис окружения.

Это неправда. Но что в наши дни . . .	\begin{Large} Это неправда. Но что в наши дни\ldots \end{Large}
--	--

Это избавит вас от подсчета множества фигурных скобок.

5.2.2. Опасность!

Как отмечено в начале этой главы, опасно усеивать ваши документы явными командами, вроде только что описанных, потому что это противоречит основной идее L^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.

Не входите в эту комнату. Она за- нята машиной неизвестного назначе- ния.	% в преамбуле или пакете \newcommand{\danger}[1]{\textbf{#1}} % в документе Не \danger{входите} в эту комнату. Она занята \danger{машиной} неизвестного назначения.
---	--

Этот подход имеет то преимущество, что вы позже можете решить, что хотите использовать другое визуальное представление опасности¹, нежели \textbf, без необходимости пробираться через весь документ, отыскивая все вхождения \textbf и определяя, отмечает ли каждое из них опасность или что-нибудь другое.

5.2.3. 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.

¹danger — прим. переводчика

5.3. Интервалы

5.3.1. 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.

5.3.2. Форматирование абзацев

Два параметра в \LaTeX 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. 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.

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.

5.3.3. Горизонтальные интервалы

\LaTeX автоматически определяет пробелы между словами и предложениями. Чтобы добавить горизонтальный пробел, пользуйтесь

```
\hspace{длина}
```

Если такой интервал должен быть выдержан, даже если он приходится на начало

¹Для добавления отступа к первому абзацу после каждого заголовка раздела пользуйтесь пакетом `indentfirst` из комплекта `'tools'`.

Дополнительный пробел между двумя строками *одного* абзаца или внутри таблицы указывается командой

```
\[длина]
```

5.4. Компоновка страницы

L^AT_EX 2_ε позволяет указать размер бумаги в команде `\documentclass`. Затем он автоматически выбирает правое поле. Но иногда predetermined значения могут вас не устроить. Безусловно, вы их можете изменить. Иллюстрация 5.2 показывает 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 L^AT_EX весьма продуман.

Безусловно, если сравнить со страницей, выданной свежеставленным MS Word, то страницы L^AT_EX 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 L^AT_EX. 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 . . .

L^AT_EX 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}
```

¹STAN:macros/latex/packages/tools

²Имея в виду настоящую печатную книгу, выпущенную уважаемым издательством.

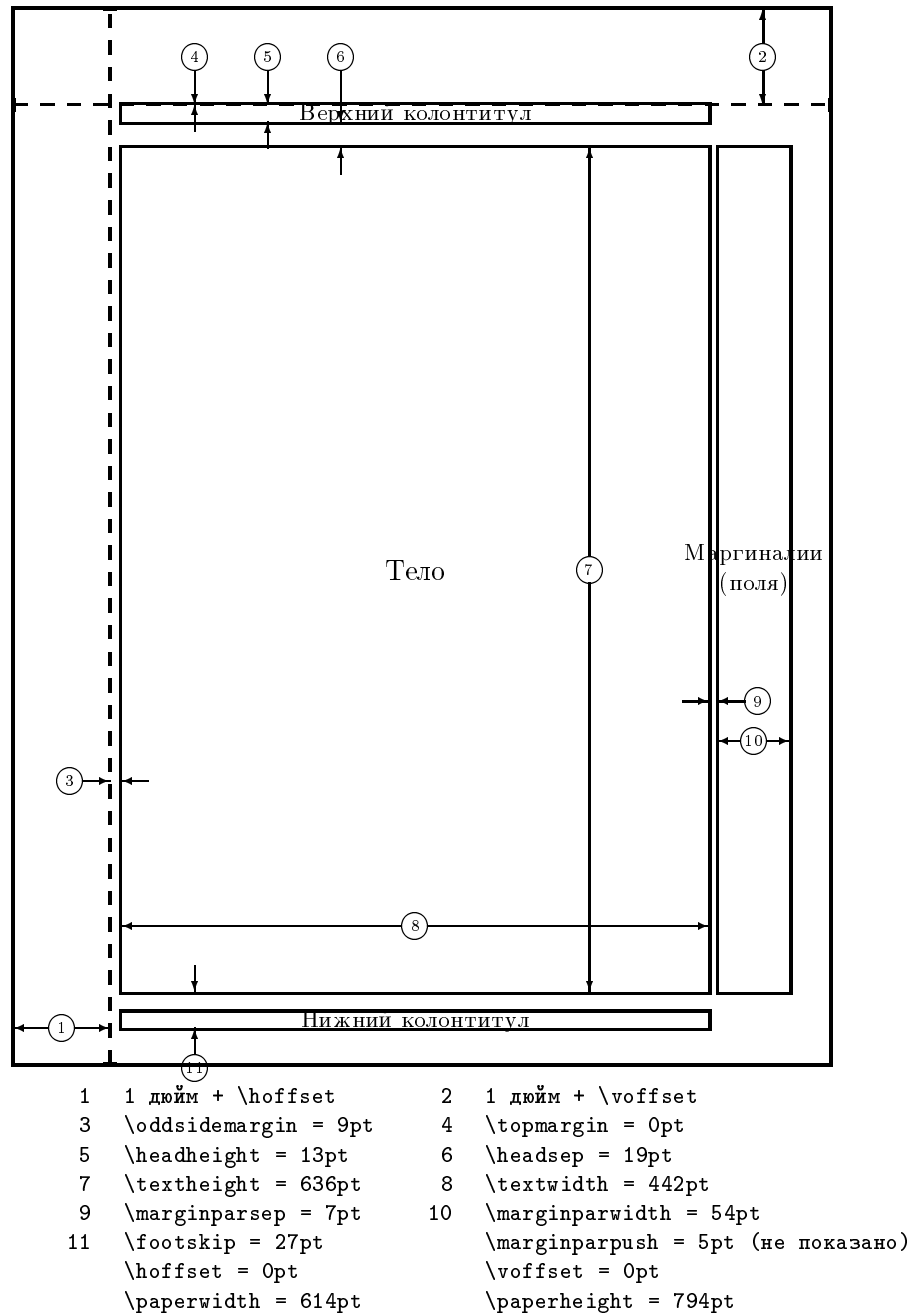


Рис. 5.2. Параметры компоновки страницы

Здесь вам может быть интересен пакет `calc`, который позволяет использовать арифметические операции в аргументе `\setlength` и в других местах, где аргументами являются численные значения.

5.5. Еще о длинах

Всегда, когда это возможно, избегайте использовать в документах абсолютных величин. Лучше основывайтесь на ширине или высоте других элементов страницы. Для ширины иллюстрации этим может служить `\textwidth`, чтобы она заполняла страницу целиком.

Следующие три команды позволяют определить ширину, высоту и глубину текстовой строки.

```
\settoheight{команда}{текст}
\settodepth{команда}{текст}
\settowidth{команда}{текст}
```

Нижеследующий пример показывает возможное применение этих команд.

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

Где: a , b – прилегают к прямому углу прямоугольного треугольника.

c – одинокая гипотенуза этого треугольника.

d – вообще тут не участвует. Вот загадка ...

```
\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}
```

5.6. 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 ЛАТ_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 строчки и прочее, существует класс блоковых 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 ЛАТ_EX 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 снаружи¹. 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. переводчика**.

центр
 р а с т я н у т ы й
 Я теперь в рамке!
 Ой, я слишком толстый
 ничего. Можете это прочитать?

```

\makebox[\textwidth]{%
  центр}\par
\makebox[\textwidth][s]{%
  растянутый}\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*.

Ааааааа кричал он, но даже
 стоящий рядом не заметил, что с ней
 случилось что-то ужасное.

```

\raisebox{0pt}[0pt][0pt]{\Large%
\textbf{Аааа\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 ней случилось
что-то ужасное.
  
```

5.7. 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.

Pittprop ...
Strut

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

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.
- [4] 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.
- [5] L^AT_EX3 Project Team. *L^AT_EX 2_ε for authors*. Включен в поставку L^AT_EX 2_ε как `usrguide.tex`.
- [6] 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`.
- [7] L^AT_EX3 Project Team. *L^AT_EX 2_ε Font selection*. Включен в поставку L^AT_EX 2_ε как `fntguide.tex`.
- [8] 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.
- [9] 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.
- [10] Graham Williams. *The TeX Catalogue* полный список множества пакетов, имеющих отношение к T_EX и L^AT_EX. Доступен в Интернет по адресу `CTAN:help/Catalogue/catalogue.html`
- [11] Keith Reckdahl. *Using EPS Graphics in L^AT_EX 2_ε Documents* объясняет все, что вы когда бы то ни было хотели знать про EPS файлы и их использование в документах L^AT_EX. Доступен в Интернет по адресу `CTAN:info/epslatex.ps`

Предметный указатель

- \! 36
- " 16
- \$ 31
- ' 34
- \(31
- \) 31
- \, 32, 36
 - 17
 - 17
- \- 16
 - 17
 - 17
 - ... 17
- \: 36
- \; 36
- \@ 20
- \[31
- \\ 15, 24–26, 63
- * 15
 - и без точек 18
 - POSTSCRIPT 49
- \] 31
- ^ 33
- _ 33
- | 26
- ~ 20

- \addtolength 63
- æ18
- amsbsy 41
- amsmath 33, 48
- amsmath 36–38, 41
- amssymb 33, 42
- \and 21
- \appendix 20
 - array 37
- \atop 35
- \author 21

- babel 6, 18, 19
- \backmatter 21
- backslash 5
- \backslash 5
- \begin 23
- \bibitem 51
- \Big 36
- \big 36
- \Bigg 36
- \bigg 36
- \bmod 35
- \boldmath 40
- \boldsymbol 41

- calc 65
- \caption 29
- \cdots 36
- center 24
- \chapter 20, 53
- \chaptermark 53
- \choose 35
- \ci 55
- \cite 51
- \cleardoublepage 30
- \clearpage 30
- command 55

- \date 21
- dcolumn 27
- \ddots 36
- \depth 66, 67
- description 23
- displaymath 31
- \displaystyle 39
- doc 10
- \documentclass 8, 15
- \dum 55

- \emph 22, 58

- empty 11
- Encapsulated POSTSCRIPT 49
- \end 23
- enumerate 23
- eqnarray 37
- equation 31, 37
- eucal 48
- eufrak 48
- exscale 10, 36

- fancyhdr 53
- figure 28, 29, 49
- flushleft 24
- flushright 24
- foiltex 8
- fontenc 10, 19
- \footnote 22
- \footnotesize 59
- \frac 35
- \framebox 66
- \frenchspacing 20
- \frontmatter 21
- \fussy 15

- GhostScript 49
- graphicx 49, 50

- textttheadings 11
- \height 66, 67
- \hline 26
- \hspace 57, 62
- \Huge 59
- \huge 59
- \hyphenation 16

- \idotsint 37
 - ifthen 10
- \iiiint 37
- \iiint 37
- \iint 37
- \include 11, 12
- \includegraphics 50, 66
- \includeonly 11, 12
- \indent 61
 - indentfirst 61
- \index 52
- \input 12
 - inputenc 10, 19
- \int 35
- \item 23
 - itemize 23
- Knuth, Donald E. 1
- \label 22, 32
 - Lamport, Leslie 1
- \LARGE 59
- \Large 59
- \large 59
 - L^AT_EX 2.09 1
 - L^AT_EX 2_ε 1
 - L^AT_EX 3 1, 4
 - latexsym 10
 - layout 63
- \ldots 17, 36
- \left 35
- \leftmark 53
- \linebreak 15
- \linespread 61
- \listoffigures 29
- \listoftables 29

- \mainmatter 21
- \makebox 66
 - makeidx 10, 52
- \makeindex 52
- \maketitle 21
 - math 31
- \mathbb 33
- \mathbf 59
- \mathcal 59
- \mathit 59
- \mathnormal 59
- \mathrm 38, 59
- \mathsf 59
- \mathtt 59
- \mbox 16, 17, 66
- \minipage 66
 - minipage 66
 - Mittelbach, Frank 1
- \multicolumn 27

- \newcommand 56
- \newenvironment 56, 57
- \newline 15
- \newpage 15

-
- `\newtheorem` 39
 - `\noindent` 61
 - `\nolinebreak` 15
 - `\nonumber` 38
 - `\nopagebreak` 15
 - `\normalsize` 59
 - œ18
 - `\overbrace` 34
 - overfull hbox 15
 - `\overleftarrow` 34
 - `\overline` 34
 - `\overrightarrow` 34
 - `\pagebreak` 15
 - `\pageref` 22
 - `\pagestyle` 11
 - `\paragraph` 20
 - `\parbox` 66
 - `\parindent` 61
 - `\parskip` 61
 - `\part` 20
 - plain 11
 - `\pmod` 35
 - `\printindex` 52
 - `\providecommand` 56
 - `\ProvidesPackage` 57
 - `\qqquad` 32, 36
 - `\quad` 32, 36
 - quotation 25
 - quote 24
 - `\raisebox` 67
 - `\ref` 22, 32
 - `\renewcommand` 56
 - `\renewenvironment` 57
 - `\right` 35
 - `\right.` 35
 - `\rightmark` 53
 - `\rule` 57, 67
 - `\scriptscriptstyle` 39
 - `\scriptsize` 59
 - `\scriptstyle` 39
 - `\section` 20, 53
 - `\sectionmark` 53
 - `\setlength` 61, 63
 - `\settodepth` 65
 - `\settoheight` 65
 - `\settowidth` 65
 - showidx 52
 - `\sloppy` 15
 - `\small` 59
 - `\sqrt` 33
 - `\stretch` 57, 62
 - `\subparagraph` 20
 - `\subsection` 20
 - `\subsectionmark` 53
 - `\subsubsection` 20
 - `\sum` 35
 - syntonly 10
 - table 28, 29, 49
 - `\tableofcontents` 21
 - tabular 26, 37, 66
 - `\textbf` 58
 - `\textit` 58
 - `\textmd` 58
 - `\textnormal` 58
 - `\textrm` 38, 58
 - `\textsc` 58
 - `\textsf` 58
 - `\textsl` 58
 - `\textstyle` 39
 - `\texttt` 58
 - `\textup` 58
 - thebibliography 51
 - `\thispagestyle` 11
 - `\tiny` 59
 - `\title` 21
 - `\tnss` 56
 - `\totalheight` 66
 - `\totalwidth` 67
 - `\underbrace` 34
 - underfull hbox 15
 - `\underline` 34
 - `\usepackage` 8, 19, 57
 - `\vdots` 36
 - `\vec` 34
 - `\verb` 25, 26
 - verbatim 54
 - verbatim 25, 54
 - `\verbatim@font` 54

- \verbatiminput 54
- verse 25
- \vspace 62
- \widehat 34
- \widetilde 34
- \width 66, 67
- WYSIWYG 3
- ажурные полужирные символы 33
- акцент
 - математический 34
- акценты 18
 - acute 18
 - grave 18
 - umlaut 18
- без засечек 58
- библиография 51
- буквы
 - европейские 18
- векторы 34
- верхние индексы 33
- входной файл 6
- выделение 22
- выравнивание
 - вправо или влево 24
 - по десятичной точке 27
- графика 8, 49
- греческие буквы 33
- группирование 60
- два столбца 9
- двусторонний вывод 9
- дефис 17
- длина 62
- длинное тире 17
- длинные уравнения 38
- дробь 35
- другие языки 18
- единицы 62
- заголовок документа 9
- запятая 17
- знак минуса 17
- иллюстрации 29
- интервал
 - двойной 61
 - междустрочный 61
- кавычки 16
- капитель 58
- квадратные скобки 5
- квадратный корень 33
- класс
 - article 8
 - book 8
 - report 8
 - slides 8
- кодировка шрифта 10
- колонтитул
 - верхний 11
 - нижний 11
- команда
 - \! 36
 - \(31
 - \) 31
 - \, 32, 36
 - \- 16
 - \: 36
 - \; 36
 - \@ 20
 - \[31
 - \\ 15, 24–26, 63
 - * 15
 - \] 31
 - \addtolength 63
 - \and 21
 - \appendix 20
 - \atop 35
 - \author 21
 - \backmatter 21
 - \backslash 5
 - \begin 23
 - \bibitem 51
 - \Big 36
 - \big 36
 - \Bigg 36
 - \bigg 36
 - \bmod 35
 - \boldmath 40
 - \boldsymbol 41
 - \caption 29

<code>\cdots</code> 36	<code>\leftmark</code> 53
<code>\chapter</code> 20, 53	<code>\linebreak</code> 15
<code>\chaptermark</code> 53	<code>\linespread</code> 61
<code>\choose</code> 35	<code>\listoffigures</code> 29
<code>\ci</code> 55	<code>\listoftables</code> 29
<code>\cite</code> 51	<code>\mainmatter</code> 21
<code>\cleardoublepage</code> 30	<code>\makebox</code> 66
<code>\clearpage</code> 30	<code>\makeindex</code> 52
<code>\date</code> 21	<code>\maketitle</code> 21
<code>\ddots</code> 36	<code>\mathbb</code> 33
<code>\depth</code> 66, 67	<code>\mathbf</code> 59
<code>\displaystyle</code> 39	<code>\mathcal</code> 59
<code>\documentclass</code> 8, 15	<code>\mathit</code> 59
<code>\dum</code> 55	<code>\mathnormal</code> 59
<code>\emph</code> 22, 58	<code>\mathrm</code> 38, 59
<code>\end</code> 23	<code>\mathsf</code> 59
<code>\footnote</code> 22	<code>\mathtt</code> 59
<code>\footnotesize</code> 59	<code>\mbox</code> 16, 17, 66
<code>\frac</code> 35	<code>\minipage</code> 66
<code>\framebox</code> 66	<code>\multicolumn</code> 27
<code>\frenchspacing</code> 20	<code>\newcommand</code> 56
<code>\frontmatter</code> 21	<code>\newenvironment</code> 56, 57
<code>\fussy</code> 15	<code>\newline</code> 15
<code>\height</code> 66, 67	<code>\newpage</code> 15
<code>\hline</code> 26	<code>\newtheorem</code> 39
<code>\hspace</code> 57, 62	<code>\noindent</code> 61
<code>\Huge</code> 59	<code>\nolinebreak</code> 15
<code>\huge</code> 59	<code>\nonumber</code> 38
<code>\hyphenation</code> 16	<code>\nopagebreak</code> 15
<code>\idotsint</code> 37	<code>\normalsize</code> 59
<code>\iiiint</code> 37	<code>\overbrace</code> 34
<code>\iiint</code> 37	<code>\overleftarrow</code> 34
<code>\iint</code> 37	<code>\overline</code> 34
<code>\include</code> 11, 12	<code>\overrightarrow</code> 34
<code>\includegraphics</code> 50, 66	<code>\pagebreak</code> 15
<code>\includeonly</code> 11, 12	<code>\pageref</code> 22
<code>\indent</code> 61	<code>\pagestyle</code> 11
<code>\index</code> 52	<code>\paragraph</code> 20
<code>\input</code> 12	<code>\parbox</code> 66
<code>\int</code> 35	<code>\parindent</code> 61
<code>\item</code> 23	<code>\parskip</code> 61
<code>\label</code> 22, 32	<code>\part</code> 20
<code>\LARGE</code> 59	<code>\pmod</code> 35
<code>\Large</code> 59	<code>\printindex</code> 52
<code>\large</code> 59	<code>\providecommand</code> 56
<code>\ldots</code> 17, 36	<code>\ProvidesPackage</code> 57
<code>\left</code> 35	<code>\qqquad</code> 32, 36

- `\quad` 32, 36
- `\raisebox` 67
- `\ref` 22, 32
- `\renewcommand` 56
- `\renewenvironment` 57
- `\right` 35
- `\right.` 35
- `\rightmark` 53
- `\rule` 57, 67
- `\scriptscriptstyle` 39
- `\scriptsize` 59
- `\scriptstyle` 39
- `\section` 20, 53
- `\sectionmark` 53
- `\setlength` 61, 63
- `\settodepth` 65
- `\settoheight` 65
- `\settowidth` 65
- `\sloppy` 15
- `\small` 59
- `\sqrt` 33
- `\stretch` 57, 62
- `\subparagraph` 20
- `\subsection` 20
- `\subsectionmark` 53
- `\subsubsection` 20
- `\sum` 35
- `\tableofcontents` 21
- `\textbf` 58
- `\textit` 58
- `\textmd` 58
- `\textnormal` 58
- `\textrm` 38, 58
- `\textsc` 58
- `\textsf` 58
- `\textsl` 58
- `\textstyle` 39
- `\texttt` 58
- `\textup` 58
- `\thispagestyle` 11
- `\tiny` 59
- `\title` 21
- `\tnss` 56
- `\totalheight` 66
- `\totalwidth` 67
- `\underbrace` 34
- `\underline` 34
- `\usepackage` 8, 19, 57
- `\vdots` 36
- `\vec` 34
- `\verb` 25, 26
- `\verbatim@font` 54
- `\verbatiminput` 54
- `\vspace` 62
- `\widehat` 34
- `\widetilde` 34
- `\width` 66, 67
- команды 5
- комментарии 6
- компоновка страницы 63
- короткое тире 17
- курсив 58
- лигатура 17
- линии
 - горизонтальные 34
- лист
 - A4 9
 - A5 9
 - B5 9
 - executive 9
 - legal 9
 - letter 9
 - титульный 9, 21
- математика 31
- математические
 - функции 34
- математический
 - минус 17
 - ограничитель 36
 - пробел 36
- многоточие 17
- наклонный 58
- немецкий язык 19
- необязательные параметры 5
- нижние индексы 33
- оглавление 21
- ограничители 35
- односторонний вывод 9
- окружение 23
 - array 37
 - center 24

- command 55
- description 23
- displaymath 31
- enumerate 23
- eqnarray 37
- equation 31, 37
- figure 28, 29, 49
- flushleft 24
- flushright 24
- itemize 23
- math 31
- minipage 66
- quotation 25
- quote 24
- table 28, 29, 49
- tabular 26, 37, 66
- thebibliography 51
- verbatim 25, 54
- verse 25
- оператор
 - интеграла 35
 - суммы 35
- опции 8
- пакет 6, 8, 55
 - amsbsy 41
 - amsmath 33, 48
 - amsmath 36–38, 41
 - amssymb 33, 42
 - babel 6, 18, 19
 - calc 65
 - dcolum 27
 - doc 10
 - eucal 48
 - eufrak 48
 - exscale 10, 36
 - fancyhdr 53
 - fontenc 10, 19
 - graphicx 49, 50
 - ifthen 10
 - indentfirst 61
 - inputenc 10, 19
 - latexsym 10
 - layout 63
 - makeidx 10, 52
 - showidx 52
 - syntonly 10
 - verbatim 54
- пакет makeidx 52
- параметр 5
- перекрестные ссылки 21
- плавающие объекты 28
- поле 63
- полужирные символы 33, 40
- полужирный 58
- правила переноса 18
- преамбула 6
- предметный указатель 52
- преимущества L^AT_EX³
- пробел 4
 - в начале строки 4
 - вертикальный 62
 - горизонтальный 62
 - после команды 5
- программа makeindex 52
- производная 34
- прямой 58
- пустые символы 4
- размер бумаги 63
- размер листа 9
- размер основного шрифта 9
- разрывы строк 15
- распорка 68
- символы
 - зарезервированные 5
- системы уравнений 37
- скобки 35
- специальные символы 18
- спецификация размещения 28
- стили страницы 11
- стиль страницы
 - empty 11
 - headings 11
 - plain 11
- стрелки 34
- структура файла 6
- таблицы 29
- тильда 34
- тильда (~) 20
- тире 17
 - длинное 17
 - короткое 17

титульный лист 21

точка 17

точка, пробел после 20

точки

 вертикальные 36

 горизонтальные 36

 диагональные 36

три точки 36

фигурные скобки 5, 58

 горизонтальные 34

формулы 31

функция модуля 35

цветной текст 8

шрифт 58

 документа, размер 9

 математический, размер 38

 размер 58, 59

