

И.А.Грицаенко, С.В.Клименко
L^AT_EX. Руководство для пользователей.
Часть I.

Предисловие

Настоящее руководство является частью технической документации по компьютерной издательской системе, основу которой составляют процессор T_EX и его макронадстройки — Plain T_EX, L^AT_EX, A_MS-T_EX. Оно содержит справочную информацию по макронадстройке L^AT_EX.

Руководство разделено на две части.

К первой части отнесены компоненты, реализующие основные концепции L^AT_EX'a — систему команд форматирования и набор файлов описания стиля, автоматизирующих подготовку общепринятых типов научных публикаций. Эта часть программного обеспечения написана самим автором L^AT_EX'a или при его непосредственном участии, является устоявшейся и будет совместимой с любыми новыми версиями L^AT_EX'a.

Ко второй части отнесены те процедуры, которые в настоящее время не являются установившимися, особенно в русскоязычной версии системы: подготовка алфавитного указателя, построение библиографии из библиографических баз данных, ввод внешней графики, коллекция шрифтов и работа с ней. Естественно, соответствующая документация непременно будет подвергаться частым изменениям, в отличие от части первой. Простота внесения изменений является принципиально важным требованием к любой технической документации, поэтому объединение столь разных по степени подготовки частей в одно громоздкое описание представляется автору неразумным.

Настоящее описание есть первая часть руководства.

Несколько слов о системе в целом.

Рассматриваемая издательская система в первую очередь предназначена для подготовки, в том числе и авторской, научных публикаций. Ориентированность на специалистов в точных и естественных науках выражается в наличии особо мощного аппарата для построения математических выражений, диаграмм, физических и химических формул, средств введения внешней графики.

Система позволяет готовить публикации на английском, русском, французском, немецком, итальянском и испанском языках с возможностью перехода с одного языка на другой в любом фрагменте текста.

Ядро системы — процессор $\text{T}_{\text{E}}\text{X}$ — представляет собой машинно-независимый язык форматирования полиграфических документов. Автор $\text{T}_{\text{E}}\text{X}$ 'а, профессор Станфордского университета США Дональд Кнут, определяет $\text{T}_{\text{E}}\text{X}$ как *“инструмент для превращения набранного компьютерным образом манускрипта в документ, полиграфическое качество которого сопоставимо с тем, что дают самые современные печатающие устройства”*.

Фундамент формирующего инструмента $\text{T}_{\text{E}}\text{X}$ 'а образуют более 300 команд-примитивов. Примитивы осуществляют операции нижнего уровня, неразложимые на более простые функциональные компоненты. На основе примитивов можно строить макрокоманды, которые могут иметь параметры и включать другие макрокоманды. Совокупность макрокоманд, подчиненных общим функциональным целям, объединяются в макронадстройки.

Первая макронадстройка $\text{T}_{\text{E}}\text{X}$ 'а, данная Д.Кнутом в качестве принятого по умолчанию формата подготавливаемых публикаций, называется Plain $\text{T}_{\text{E}}\text{X}$ и содержит приблизительно 600 макрокоманд. Эти макрокоманды определяют типы и размеры шрифтов заголовков и основного текста, высоту и ширину страниц, промежутки между строками и между абзацами, отступ на первой строке абзаца, тип нумерации страниц (числа римские или арабские) и многое другое.

Формат Plain $\text{T}_{\text{E}}\text{X}$ есть стартовая точка для построения практически всех новых макронадстроек. Большинство его макроопределений настолько фундаментальны, что как правило включаются в другие макронадстройки без каких-либо модификаций.

$\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ — макронадстройка $\text{T}_{\text{E}}\text{X}$ 'а, предназначенная для *автоматизации* подготовки к публикации научно-технических документов стандартных типов. К стандартным типам документов относятся статьи, доклады, книги и деловые письма.

Первая версия $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 'а появилась в 1985 году. Автор ее — Лесли Лампорт.

Фундаментальной концепцией $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 'а является идея *стиля документа*, которая реализуется выбором надлежащего *базового стиля* и группы *корректирующих стилей*, дополняющих базовый.

Система команд $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 'а имеет форматы, отличные от форматов команд $\text{T}_{\text{E}}\text{X}$ 'а. Команды $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 'а более стандартизованы и поэтому проще в использовании. Большинство команд реализует цельные процедуры верстки, позволяя пользователю не вникать в весьма непростые детали этих процедур.

Еще одной важной макронадстройкой является $\mathcal{A}\mathcal{M}\mathcal{S}\text{-T}_{\text{E}}\text{X}$. Это программное обеспечение разработано и поддерживается Американским математическим обществом. Оно содержит средства подготовки публикаций в стандартах, принятых в американских математических журналах.

$\mathcal{A}\mathcal{M}\mathcal{S}\text{-T}_{\text{E}}\text{X}$ располагает великолепной коллекцией шрифтов, в том числе кириллических с своеобразными лигатурами. Украшением этой коллекции служат математиче-

ские шрифты, уникальные по количеству и разнообразию представленных в них символов.

Эта макронадстройка имеет вариант, совместимый с макронадстройкой L^AT_EX.

Таким образом, именно L^AT_EX, упрощающий пользователю работу по подготовке публикаций и совместимый со всеми другими компонентами системы, является важнейшей частью рассматриваемой компьютерной издательской системы.

Данное пособие есть справочное руководство по макронадстройке L^AT_EX. Оно позволяет пользователю работать с версиями L^AT_EX 2.09, L^AT_EX/2e и вводимой в эксплуатацию версией L^AT_EX 3.

Руководство включает:

- общую характеристику L^AT_EX'а;
- схему получения документа с помощью L^AT_EX'а;
- описание системы команд;
- описание параметров стиля — внутренних команды L^AT_EX'а, предназначенных для построения файлов *описания стиля*, которые пользователь может переопределять в собственном документе.
- список сообщений об ошибках;
- список терминов.

Описание системы команд разделяется на подразделы, информация в которых объединена по функциональному сходству.

Определения команд набраны равноширинным шрифтом, воспроизводящим шрифт клавиатуры дисплея. Для выделения в тексте строк с определениями команд используется указующий символ ▷.

Если для какой-либо команды L^AT_EX'а имеется аналогичная команда Plain T_EX, распознаваемая L^AT_EX'ом и удобная в использовании, она также будет приведена.

Каждый раздел содержит примеры.

Разнообразие средств форматирования не позволило построить все примеры однотипным образом. Общим в представлении примеров являются начало и конец их, данные в виде длинных горизонтальных скобок, а также то, что сначала показывается или обсуждается сформатированный фрагмент, а затем приводится его исходный текст.

Следует подчеркнуть, что в данном руководстве *не рассматриваются* так называемые внутренние команды, т.е. те, которые вводятся для построения системы команд или команд в файлах описания стилей, и недопустимы в текстах подлежащих обработке документов.

Описание библиотеки файлов *корректирующих стилей* ввиду ее разнообразия и обширности составляет отдельный раздел технической документации системы.

1. Общая характеристика $\text{\LaTeX}$ 'а

$\text{\LaTeX}$ — макронадстройка $\text{\TeX}$ 'а, предназначенная для *автоматизации* подготовки к публикации научно-технических документов стандартных типов. К стандартным типам документов относятся статьи, доклады, книги и деловые письма.

$\text{\LaTeX}$ имеет собственную систему команд, формат которых отличается от форматов команд $\text{\TeX}$ 'а.

Фундаментальной концепцией $\text{\LaTeX}$ 'а, является идея *стиля документа*.

1.1. Что такое стиль документа

Стиль документа — это набор макрокоманд, определяющий полиграфические форматы будущей публикации: размер страниц, вид титульной информации, типы рубрик, типы шрифтов для заголовков различных рубрик и подразбук, правила формирования абзаца, списков с перечислением, подписей к рисункам и таблицам, оглавления, правила построения и нумерации математических формул и теорем и т.д.

Этот набор макрокоманд хранится в файле *описания стиля*.

Стиль выбирается первой командой в исходном файле документа. Обязательный параметр этой команды задает основной, *базовый* стиль документа.

Необязательным параметром можно указать несколько файлов с *корректирующими* стилями, вносящими дополнения в базовый стиль. С помощью корректирующих стилей можно, например, выбрать отличный от принятого по умолчанию размер основного шрифта документа, указать количество колонок текста на странице и многое другое.

Таким образом, первая же команда документа, обрабатываемого $\text{\LaTeX}$ 'ом, заставляет работать на документ большие массивы квалифицированно написанных макрокоманд.

1.2. Основные возможности $\text{\LaTeX}$ 'а

Здесь дан список возможностей $\text{\LaTeX}$ 'а, которыми располагает любая конфигурация системы, например, с минимальным набором файлов корректирующих стилей.

- $\text{\LaTeX}$ автоматизирует нумерацию рубрик, рисунков, таблиц, расстановку ссылок на литературу и перекрестных ссылок.
- $\text{\LaTeX}$ упрощает подготовку списка литературы, позволяя при этом использовать библиографические базы данных.
- $\text{\LaTeX}$ автоматизирует подготовку алфавитного указателя.
- $\text{\LaTeX}$ упрощает построение таблиц, в том числе таких, размер которых превышает размер страницы.
- Математические выражения строятся с автоматическим учетом принятых в полиграфии пробелов между символами в формулах, переменные по умолчанию набираются математическим курсивом, стандартные функции прямым шрифтом.

Расположение индексов в формулах со знаками предела, суммы или интеграла зависит от того, печатаются ли они отдельной строкой или в потоке текста. Строго выдержано расположение и относительные размеры символов в выражениях с верхними и нижними индексами, в том числе двойными и тройными.

- В $\text{\LaTeX}$ ’е имеется простая и красивая процедура перечислений. В зависимости от требований пользователя пункты списка нумеруются или отмечаются специальными символами или отступами, не совпадающими с абзацными.
- $\text{\LaTeX}$ позволяет ввести на странице несколько мини-страниц, причем на каждой из них (и на большой странице тоже) допустимы сноски.
- $\text{\LaTeX}$ располагает разнообразными средствами введения в документы диаграмм и рисунков.

1.3. Совместимость $\text{\LaTeX}$ ’а с другими форматирующими компонентами системы

- $\text{\LaTeX}$ распознает все примитивы $\text{\TeX}$ ’а.
- $\text{\LaTeX}$ распознает большинство команд Plain $\text{\TeX}$.
Исключение составляют команды определения стиля страницы, введения сносок и рисунков, команды табулирования, выбора некоторых шрифтов и команды выравнивания уравнений. Для всех этих операций $\text{\LaTeX}$ имеет свои наборы команд.

Команды Plain $\text{\TeX}$, не воспринимаемые $\text{\LaTeX}$ ’ом

- Команды табулирования.

<code>\tabs</code>	<code>\tabsdone</code>	<code>\settabs</code>
<code>\tabset</code>	<code>\cleartabs</code>	<code>\tabalign</code>

- Команды ввода номеров страниц, сносок и рисунков.

<code>\pageno</code>	<code>\nopagenumbers</code>	<code>\makeheadline</code>	<code>\footstrut</code>
<code>\headline</code>	<code>\advancepageno</code>	<code>\makefootline</code>	<code>\topins</code>
<code>\footline</code>	<code>\nopagenumbers</code>	<code>\dosupereject</code>	<code>\topinsert</code>
<code>\normalbottom</code>	<code>\plainoutput</code>	<code>\pagecontents</code>	<code>\midinsert</code>
<code>\folio</code>	<code>\pagebody</code>	<code>\vfootnote</code>	<code>\pageinsert</code>
			<code>\endinsert</code>

- Команды выбора шрифта.

<code>\fivei</code>	<code>\fivebf</code>	<code>\sevensy</code>
<code>\fiverm</code>	<code>\seveni</code>	<code>\teni</code>
<code>\fivesy</code>	<code>\sevenbf</code>	<code>\oldstyle</code>

- Команды выравнивания уравнений.

<code>\eqalign</code>	<code>\eqalignno</code>	<code>\leqalignno</code>
-----------------------	-------------------------	--------------------------

- Использование в командах описания стиля документа файлов корректирующих стилей, активизирующих макронастройку $\text{\LaTeX}$ делает доступными для пользователя все возможности этой макронастройки.

2. Схема получения документа с помощью L^AT_EX'a

Основной входной файл для L^AT_EX'a — это файл с исходным текстом документа и разметкой его командами. Исходный файл набирается любым редактором и как правило имеет расширение `tex`.

Запуск L^AT_EX'a в работу осуществляется командой вида

```
LATEX <имя исходного файла>
```

Если исходный файл имеет расширение `tex`, достаточно указать краткое имя файла.

В начале работы L^AT_EX читает файлы, формирующие стиль документа. Если какой-либо из стилевых файлов недоступен, L^AT_EX остановится с диагностикой, указывающей отсутствующий файл, и предложит ввести имя файла, позволяя таким образом пользователю оперативно исправить допущенную ошибку.

Точно также L^AT_EX поступит в случае, когда окажется недоступным указанный в исходном файле файл с фрагментом обрабатываемого текста.

В процессе работы L^AT_EX, как и T_EX, использует так называемый формат.

Формат — это файл, содержащий систему команд в откомпилированной форме.

Формат-файл имеет расширение `fmt`.

Для форматирования текста L^AT_EX'у необходимо знать метрику шрифтов.

Метрика шрифта — это набор чисел, определяющий для каждого графического представления символа — глифа — необходимое ему пространство на странице: размер бокса, заключающего в себя глиф, и некоторые граничные условия, например, относительные расстояния между парами глифов.

Метрика каждого шрифта хранится в файле с именем шрифта и расширением `tfm`.

В результате работы L^AT_EX'a создается несколько файлов; все они имеют имя исходного файла, но различные расширения.

Основной выходной файл L^AT_EX'a имеет расширение `dvi` (DeVice Independent file) и содержит сформатированный текст документа в машинно-независимом виде.

Для выдачи сформатированного текста на печать или экран дисплея существуют программы, называемые *драйверами*.

Если документ содержит оглавление, список литературы и ссылки на него, список рисунков и(или) список таблиц, алфавитный указатель, то L^AT_EX должен отработать дважды. В первый раз будут создаваться соответствующие рабочие файлы с расширениями:

`toc` — для оглавления;
`aux` — для ссылок;
`lof` — для списка рисунков;
`lot` — для списка таблиц;
`idx` — для списка терминов, включенных в алфавитный указатель.

Во второй раз $\text{\LaTeX}$ вставит в текст документа все ссылки и дополнит его требуемыми списками литературы, таблиц и т.д.

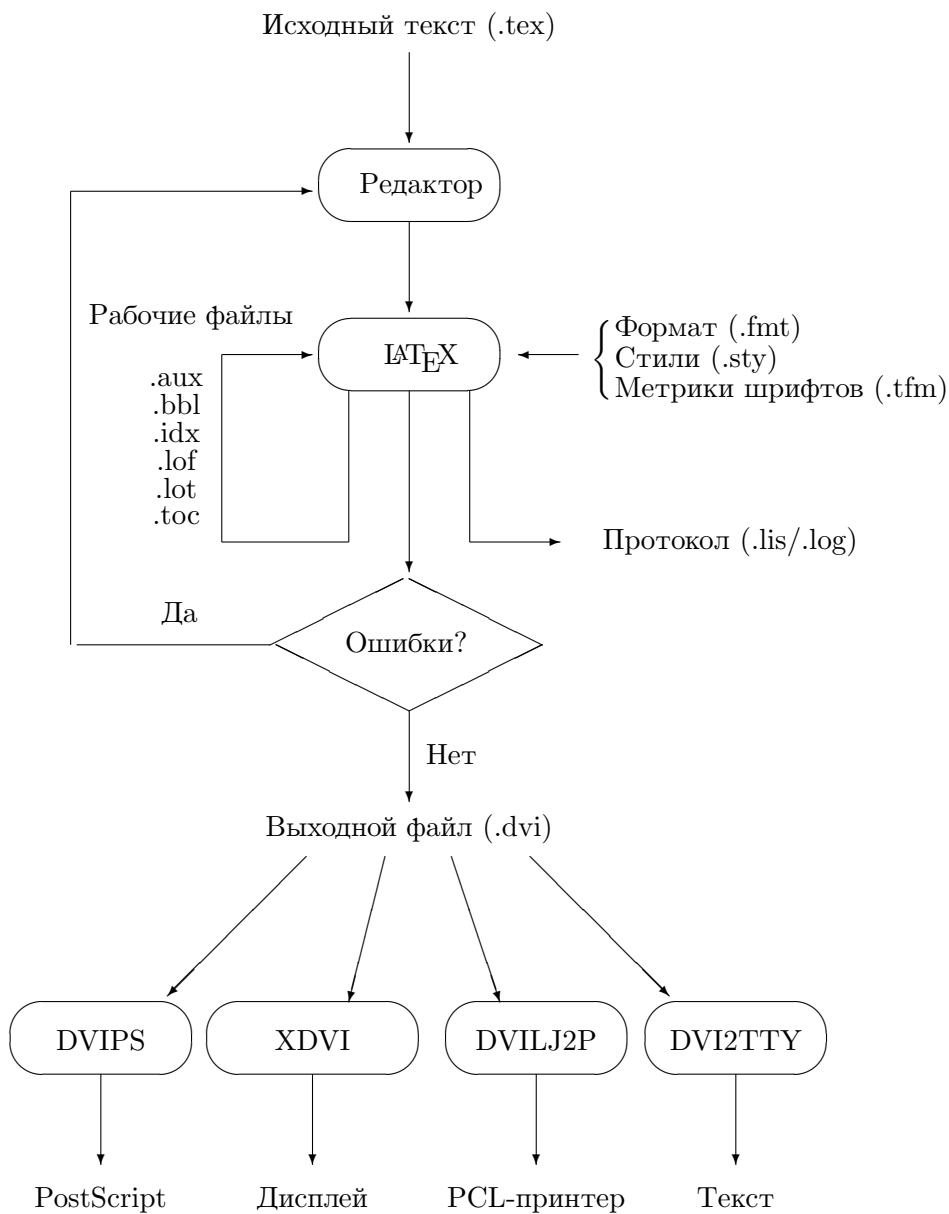


Рис. 1. Схема обработки документа $\text{\LaTeX}$ 'ом

В сложных случаях, например, когда оглавление помещено в начале документа, а расстановка перекрестных ссылок приводит к перемещению заголовка какой-либо рубрики на другую страницу, правильный документ получается лишь после трехкратной обработки текста $\text{\LaTeX}$ ’ом.

Следует отметить, что новые версии рабочих файлов создаются при каждом прогоне $\text{\LaTeX}$ ’а. При отсутствии любого из вышеуказанных рабочих файлов $\text{\LaTeX}$ выдает соответствующее сообщение и продолжает работу.

Для выдачи сформатированного текста на печать или экран дисплея существуют программы, называемые *драйверами*.

Потоки данных в цикле обработки документа представлены на рис. 1.

При обнаружении ошибки в процессе обработки текста $\text{\LaTeX}$ выдает сообщение о ней и приостанавливает работу, выдавая знак вопроса.

Возможные действия пользователя:

- Нажатие клавиши `<RETURN>` возобновит работу $\text{\LaTeX}$ ’а вплоть до следующей ошибки.
- Нажатие клавиш `<S><RETURN>` возобновит работу $\text{\LaTeX}$ ’а в ином режиме: сообщения об ошибках будут выдаваться без приостановки дальнейшей обработки исходного файла.
- Нажатие клавиш `<X><RETURN>` приведет к закрытию всех рабочих файлов и полной остановке работы с выходом на уровень операционной системы.

$\text{\TeX}$ и $\text{\LaTeX}$ могут выдавать предупреждающие сообщения, не приостанавливая работы.

Форматы сообщений об ошибках слегка различаются в зависимости от того, кто — $\text{\LaTeX}$ или $\text{\TeX}$ — диагностирует ошибку.

Сообщения об ошибках и предупреждающие сообщения рассмотрены в разделе “Диагностика ошибок” нашего руководства.

Иногда $\text{\LaTeX}$ может остановиться без всякой диагностики. Обычно это бывает при отсутствии команды конца документа или ошибке в ее формате, но, к сожалению, и другие, “неожиданные” для $\text{\LaTeX}$ ’а ошибки могут вызвать такую же ситуацию.

Для того, чтобы прервать работу $\text{\LaTeX}$ ’а, закрыв все рабочие файлы, следует ввести `<control-C>`. $\text{\LaTeX}$ остановится, как при обнаружении им ошибки. Последующее нажатие клавиш `<X><Return>` выведет на уровень операционной системы.

3. Форматы команд L^AT_EX'а

Все управляющие инструкции L^AT_EX'а можно разделить на две большие группы: инструкции, доступные пользователю, и внутренние команды.

Инструкции, доступные пользователю, могут быть использованы и переопределены в исходном файле документа. Описание их составляет главную цель данного руководства.

Внутренние команды предназначены для построения системы пользовательских команд. Они имеют в названии специальный символ @, и не могут быть переопределены. В данном руководстве внутренние команды L^AT_EX'а не рассматриваются.

Инструкции, доступные пользователю, подразделяются на три группы:

управляющие символы команды командные скобки

Инструкции двух последних типов могут содержать параметры. Основными видами параметров команд и командных скобок являются *текст*, *безразмерная величина*, *линейная длина*.

L^AT_EX различает прописные и строчные буквы. Вводите все команды строчными буквами, если не указано иначе.

3.1. Единицы измерения

В командах L^AT_EX'а длины могут быть заданы жестко или быть эластичными, т.е. обладать способностью сжиматься и растягиваться.

Распознаваемые T_EX'ом и L^AT_EX'ом единицы измерения жесткой длины и соотношения между ними приведены в таблице 1.

Таблица 1. Единицы измерения

Аббре- виатура	Название	Соотношение
pt	пункт	
pc	пика	1pc=12pt
in	дюйм	1in=72.27pt
bp	большой пункт	72bp=1in
cm	сантиметр	1in=2.54cm
mm	миллиметр	10mm=1cm
dd	пункт дидо	1157dd=1238pt
cc	цицero	1cc=12dd
sp	масштабированный пункт	65536sp=216sp=1pt
em	ширина 'М'	
ex	высота 'х'	

—

Единица измерения ‘em’ соответствует ширине заглавной буквы ‘M’ и зависит от выбранного шрифта. Единица измерения ‘ex’ есть высота в данном шрифте строчной буквы ‘x’.

Число или числовое значение линейной величины может быть указано либо как целое, либо как число с десятичной точкой. Типографский пункт принят в некотором смысле основной единицей, поскольку размеры всех шрифтов указываются в пунктах.

Если вы не укажете размерность там, где ожидается линейная величина, будет выдано сообщение об ошибке.

3.2. Управляющие символы

TeX и L^AT_EX имеют общий набор из 10 управляющих символов, выполняющих следующие функции.

- \ Признак начала команды.
- { } Признаки начала и конца группы и ограничители обязательных параметров команд. TeX и L^AT_EX имеют широкие возможности группировать материал, который в некотором смысле должен трактоваться одинаково. Например, фрагмент текста надо выделить, напечатав его иным шрифтом или центрированно, вынести его в виде сноски и прочее. Такой фрагмент оформляется в группу при помощи фигурных скобок, которые указывают область действия текущей команды. Группирование может быть распространено на любую глубину (скобки внутри скобок внутри скобок. . .) с одним только условием: скобки должны быть сбалансированы.
- ~ Неразрываемый пробел. Слова, связанные таким пробелом, не могут быть разнесены на отдельные строки.
- % Признак начала комментария. Весь текст после этого символа до конца строки игнорируется.
- # Признак параметра в TeX-определении макрокоманды.
- & Табулятор. С помощью этого символа выполняется разметка столбцов в таблицах или массивах данных.
- \$ Переключатель в математическую моду.
- ^ В математической моде признак верхнего индекса.
- _ В математической моде признак нижнего индекса.

3.3. Команды

Команды L^AT_EX’а начинаются управляющим символом \ и имеют следующий формат:

\<имя> [<необязательные параметры>]{<обязательные параметры>}

В именах команд не используются цифры.

Необязательные параметры следует помещать в квадратные скобки, обязательные — в фигурные.

<code>\it</code>	% Команда выбора шрифта для печати курсивом
<code>\section{Шрифты}</code>	% Команда ввода заголовка рубрики
<code>\rule[+3mm]{2cm}{1mm}</code>	% Команда ввода горизонтальной линии длиной 2см, % толщиной 1мм, поднятой над текущей строкой на 3мм

Существуют команды, **<имя>** которых состоит из одного символа.

<code>\\</code>	% Команда перехода на новую строку
<code>\%</code>	% Команда ввода знака процента
<code>\S</code>	% Команда ввода знака параграфа

Параметры задаются последовательностью символов, числом или размерной величиной, определяющей длину.

Характерным обязательным параметром множества команд является **<текст>**, который может содержать в себе и команды форматирования.

Число или длина могут быть заданы непосредственно или выражены через выданные ранее команды с численными параметрами,

Порядок расположения параметров команды **существенен**.

Конец команд, содержащих параметры, распознается по скобкам, обрамляющим последний параметр.

Признаком конца команды без параметров является первый встреченный символ *не буква*. Этот символ будет воспринят как начало фрагмента текста, который следует за командой. Если это пробел, то он игнорируется, и в этом случае вам может понадобиться вставка *обязательного пробела*.

Если выдается несколько команд подряд, разделителем служит символ `\`.

3.4. Командные скобки

Командные скобки предназначены для форматирования определенным образом фрагмента текста, расположенного в пределах этих скобок.

Открытие и закрытие командных скобок определяется ключевыми словами **begin** и **end**. Каждому ключевому слову предшествует символ `\`:

```
\begin{<имя>}[<необязательные параметры>]{<обязательные параметры>}
  <текст>
\end{<имя>}
```

В именах командных скобок не используются цифры.

Необязательные параметры следует помещать в квадратные скобки, обязательные — в фигурные.

```
\begin{center}          % Командные скобки центрирования текста
<текст>
\end{center}
\begin{minipage}{10cm}  % Командные скобки мини-страницы шириной в 10см
<текст>
\end{minipage}
```

Порядок расположения параметров командных скобок **существенен**.

Параметры задаются последовательностью символов, числом или размерной величиной. Число или длина могут быть заданы непосредственно или выражены через выданные ранее команды с численными параметрами,

Как правило, командные скобки образуют законченный абзац или серию абзацев, неким образом выделенных в тексте, и фрагмент текста, в них заключенный, будет отделен от остального текста дополнительными вертикальными промежутками, определяемыми в файле описания стиля.

Конец командных скобок задается стандартно командой `\end{<имя>}`.

3.5. Формат команд $\TeX$ 'а

В нашем руководстве обсуждаются команды и примитивы $\TeX$ 'а, часто используемые в документах, обрабатываемых $\LaTeX$ 'ом (по причинам историческим или в силу удобства). Формат команд $\TeX$ 'а имеют и параметры стиля — команды $\LaTeX$ 'а, используемые для построения файлов *описания стиля*, но которые пользователь может переопределять в собственном документе.

Поэтому читателям нашего руководства важно иметь общую информацию о формате команд $\TeX$ 'а.

Формат команд $\TeX$ 'а следующий.

Команды начинаются ключевым символом `\` и имеют имена, состоящие только из букв. За именем команды могут следовать аргументы и параметры.

Аргумент задается или *числом*, или *последовательностью символов*. Если в команде требуется числовой аргумент, он приравнивается имени команды. Знак равенства не является обязательным, его обычно включают для наглядности.

Число может быть выражено безразмерной или размерной величиной или командой, которой уже присвоено некоторое значение.

Не все аргументы являются обязательными.

Параметром команды как правило является фрагмент текста.

Признаком конца команды без аргументов и параметров является *не буква*. Этот символ будет воспринят как начало фрагмента текста, следующего за командой. Если

—
это пробел, то он игнорируется, и в этом случае может понадобиться вставка *обязательного пробела*.

Если выдается несколько команд подряд, разделителем служит символ \. Т_EX различает строчные и прописные буквы.

<code>\par</code>	% Команда начала абзаца
<code>\vskip 2cm</code>	% Команда ввода вертикального промежутка в 2см
<code>\fbox{шрифты}</code>	% Команда обведения рамкой указанного текста
<code>\centerline{<текст>}</code>	% Команда центрирования указанного текста
<code>\hskip 0.5\hsize</code>	% Команда ввода горизонтального промежутка, % равного половине ширины страницы.

4. Некоторые сведения о стилях

К базовым стилям оригинального L^AT_EX'а относятся:

- `article` — статья;
- `report` — доклад;
- `book` — книга;
- `letter` — письмо.

Практика показывает, что первые три стиля, сходные по многим форматам, используются у нас чрезвычайно интенсивно, тогда как форма делового письма, принятая в России и на Западе (расположение адресов, даты, подписей), отличается настолько сильно, что делает последний базовый стиль в его оригинальной форме малоупотребительным.

Здесь мы опишем кратко, что же именно определяется в базовых стилях «статья», «доклад» и «книга».

Прежде всего определим понятие основного размера шрифта документа.

Основным размером шрифта документа принято считать размер, которым напечатана подавляющая часть его фрагментов, и который таким образом является одной из опорных характеристик, определяющих зрительное восприятие документа.

Выбор основного размера определяет остальные 9 размеров для шрифтов, автоматически активизируемых L^AT_EX'ом: пять размеров больше основного и два меньше основного, используемых для формирования заголовков рубрик различных уровней, подписей к рисункам и таблицам и других целей, а также размеры шрифтов для сносок и для индексов в математических выражениях.

Через основной размер шрифта вычисляются расстояния между строками, между абзацами, между заголовками и последующими фрагментами текста, размеры полей, отступ первой сноски от основного текста и множество других параметров пространственного формирования страницы.

Задание основного размера шрифта документа и вычисление связанных с ним параметров образует «переменную» часть базовых стилей. Такие части вынесены в отдельные файлы *корректирующих стилей* с названиями

`art10` `art11` `art12`

Эти файлы, первые в обширной библиотеке корректирующих стилей, определяют в качестве основного размера шрифта соответственно 10pt, 11pt, 12pt.

В команде описания стиля документа допустима краткая форма вызова этих стилей, именно:

`10pt` `11pt` `12pt`

Если в списке корректирующих стилей не задан ни один из них, по умолчанию будет использован стиль 10pt.

При исполнении команд файла базового стиля прежде всего читается надлежащий файл с заданием основного размера шрифта документа. Данные его используются при определении большинства команд форматирования материала в соответствии с полиграфическими требованиями в данному типу публикации.

Базовый стиль, в частности, определяет:

- Расположение титульной информации. Заголовок и сведения об авторах статьи печатаются над началом текста, тогда как в стилях «доклад» и «книга» они вынесены на отдельную страницу.
- Типы рубрик и их форматы, а также уровень глубины их вложения. Стил **article** допускает рубрики «часть», «секция» и далее 2 рубрики младше секции (подсекция и подподсекция). Стили **report** и **book** допускают также рубрику «глава», старшую по отношению к рубрике «секция». Разрешенная глубина вложения рубрик равна 4. Это значит, что если в докладе вы начали с рубрики «часть», то получить автоматически нумеруемую рубрику «подподсекция» (пятый уровень вложения) невозможно.
- Местоположение, тип и размер шрифта для печати бегущего заголовка (колоннотитула). Бегущим заголовком называют заголовок раздела, появляющийся на каждой странице этого раздела.
- Форматы подписей к рисункам и таблицам.
- Командные скобки построения списков с различными типами меток для отдельных пунктов списка (число, специальный символ, особый отступ) и глубину их вложения, равную 4.
- Форматы сносок (т.е. замечаний, снабженных метками и располагаемых ниже основного текста на странице) и заметок на полях.
- Вид и расположение номеров математических выражений при автоматической их нумерации.
- Форматы списков литературы, таблиц, рисунков.
- Форматы перекрестных ссылок (т.е. ссылок на предыдущие разделы работы, нумерованные математические выражения и прочее) и ссылок на литературу.
- Команды формирования титульной страницы и аннотации.

Коллекция корректирующих стилей огромна. Она рассматривается в отдельном томе нашей технической документации.

5. Команды задания стиля, начала и конца документа

Рассматриваемые здесь команды имеют общую особенность: они являются обязательными для любого документа и воздействуют на документ в целом. Команда описания стиля и командные скобки начала и конца документа могут быть заданы только один раз, и положение их в исходном тексте является фиксированным.

Исходный файл, содержащий размеченный командами текст документа, логически подразделяется на две части — преамбулу и тело документа.

Преамбула содержит команды описательного характера, не исполняемые непосредственно, а уточняющие форматы будущего документа или вводящие новые средства форматирования в виде макрокоманд. Именно в преамбуле удобно изменять параметры стиля.

Вторая часть есть собственно предмет воздействия $\text{\LaTeX}$ 'а после прочтения им информации из всех файлов описания стилей и преамбулы. Эта часть документа является текстом, подлежащим обработке.

Задания стиля документа в $\text{\LaTeX}$ 2.09 отличается от задания стиля в $\text{\LaTeX}$ /2 ϵ и $\text{\LaTeX}$ 3.

5.1. Задание стиля в $\text{\LaTeX}$ 2.09

▷ `\documentstyle[<корректирующие стили>] {<базовый стиль>}`

Команда определяет необходимые файлы описания стилей и должна быть первой командой документа. Обязательный параметр должен быть именем одного из базовых стилей, т.е. стандартных стилей научно-технических документов, автоматизацию подготовки которых декларирует $\text{\LaTeX}$.

Файлы описания стилей, базовых или корректирующих, имеют в $\text{\LaTeX}$ 2.09 расширение `sty`. В команде описания стиля имена файлов указываются без расширения.

К базовым стилям $\text{\LaTeX}$ 'а относятся:

`article` `report` `book` `letter`

В качестве необязательных параметров в команде могут быть указаны один или несколько корректирующих стилей. Если вы используете более одного корректирующего стиля, их имена должны быть разделены запятой. Так как $\text{\LaTeX}$ читает файлы корректирующих стилей последовательно, ограничений на их количество нет.

5.2. Задание стиля в $\text{\LaTeX}$ /2 ϵ и $\text{\LaTeX}$ 3

Главная цель создания новой версии состояла в построении оптимального и эффективного интерфейса с пакетами файлов корректирующих стилей. Потребность в таком интерфейсе можно назвать насущной ввиду огромного количества файлов корректирующих стилей, накопленного к настоящему моменту.

Ранее такой интерфейс строился только посредством указания списка файлов корректирующих стилей в команде задания стиля документа, никакого формального упорядочивания этих файлов не было предусмотрено.

В L^AT_EX/2_ε и L^AT_EX 3 понятие базовых стилей трансформируется в понятие *стандартный класс документа*. Таких классов пять:

article report book slides letter

Файлы описания классов имеют расширение `cls`.

Файлы корректирующих стилей полностью сохраняют формат предыдущей версии и имеют расширение `sty`.

Команда задания стиля документа имеет тот же формат, что и в L^AT_EX 2.09:

▷ `\documentstyle[<корректирующие стили>] {<класс>}`

Обязательный параметр должен быть именем одного из стандартных классов.

В качестве необязательных параметров в команде могут быть указаны один или несколько корректирующих стилей. Если вы используете более одного корректирующего стиля, их имена должны быть разделены запятой.

В преамбулу документа рекомендуется вводить команды `\usepackage`. Каждая такая команда декларирует определенный пакет стилевых файлов, который в дальнейшем будет использоваться. Формат ее следующий:

▷ `\usepackage[<параметры выбора>] {<пакет>}`

Обязательный параметр указывает имя пакета стилевых файлов. Если он допускает различные варианты работы, в параметрах выбора можно указать только те, которые полезны при обработке вашего документа.

5.3. Указание начала и конца обрабатываемого текста

Командные скобки начала и конца текста документа есть

▷ `\begin{document}`
 <тело документа>
`\end{document}`

Начало командных скобок `\begin{document}` отмечает конец преамбулы и начало подлежащего обработке текста. Конец командных скобок `\end{document}` считается последней строкой документа.

6. Выбор языка документа

▷ `\selectlanguage{<язык документа>}`

Команда выбора языка документа, который будет использоваться вплоть до конца документа или до следующей команды `\selectlanguage`.

Требуемый выбор языков должен быть указан надлежащими файлами корректирующих стилей в команде `\documentstyle`:

<code>english</code>	— английский;
<code>russian</code>	— русский;
<code>germanb</code>	— немецкий;
<code>francais</code>	— французский;
<code>italian</code>	— итальянский;
<code>spanish</code>	— испанский.

Значение обязательного параметра `<язык документа>` совпадает с названием соответствующего корректирующего стиля.

По данной команде включаются надлежащие таблицы образцов переносов слов и служебных слов («Содержание», «Алфавитный указатель» и прочее), а также устанавливаются некоторые стандарты набора. Например, в публикациях на французском языке после знаков пунктуации используются иные пробелы, нежели в публикациях на английском.

7. Ввод простого текста

В данном разделе представлены средства ввода и форматирования простого текста — без математических выражений, таблиц, диаграмм, построенных командами $\text{\LaTeX}$, и внешней графики.

При обработке команд и командных скобок, описанных здесь, $\text{\LaTeX}$ автоматически строит строки, абзацы и страницы, определяет места переносов слов и варьирует расстояния между словами так, чтобы вид строки соответствовал полиграфическим требованиям. Такая мода работы называется абзацной.

7.1. Символы, вводимые «как есть»

Как уже упоминалось, исходный файл документа, подготовляемого к публикации средствами $\text{\LaTeX}$, содержит информацию двух типов: текст и разметку (команды форматирования). Текст в основном вводится «как есть», хотя существуют символы, вводимые особым образом, либо потому, что их нет на клавиатуре вводного устройства, либо потому, что $\text{\LaTeX}$ использует их как управляющие символы.

Путем простого нажатия соответствующей клавиши дисплея вводятся:

- Строчные буквы.
a b c d e f g h i j k l m n o p q r s t u v w x y z
а б в г д е е ж з и й к л м н о п р с т у ф х ц ч ш щ ь ы э ю я
- Прописные буквы.
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
А Б В Г Д Е Е Ж З И Й К Л М Н О П Р С Т У Ф Х Ц Ч Ш Щ Ъ Ы Э Ю Я
- Арабские цифры.
0 1 2 3 4 5 6 7 8 9
- Знаки пунктуации, скобки и иные небуквенные символы, имеющиеся на клавиатуре любого дисплея.
. , ? ! : ; () [] ‘ ’ - * / + = @
- Пробелы, табуляторы и признак конца строки.

7.2. Пробелы

Ввод с клавиатуры нескольких пробелов $\text{\LaTeX}$ рассматривает как один пробел и свободно сужает или расширяет его, следя за тем, чтобы вид строки соответствовал полиграфическим требованиям.

Пробел, которому предшествует символ $\backslash$, является обязательным.

Пробел, вводимый символом $\sim$, является неразрываемым и всегда помещает связываемые им слова в одну строку.

Ввод конца строки интерпретируется как пробел. Если это нежелательно, перед вводом конца строки следует задать символ комментария $\%$.

Для вставки пробелов различных размеров существуют следующие команды.

- ▷ `\thinspace`
Задает пробел наименьшего допустимого размера.
- ▷ `\enspace`
Задает пробел, равный половине высоты заглавных букв.
- ▷ `\quad`
Задает пробел, равный высоте заглавных букв.
- ▷ `\qquad`
Задает пробел, равный двойной высоте заглавных букв.

! пробел самый маленький	<code>!\thinspace</code> пробел самый маленький <code>\par</code>
! пробел чуть больший	<code>!\enspace</code> пробел чуть больший <code>\par</code>
! пробел еще больший	<code>!\quad</code> пробел еще больший <code>\par</code>
! пробел наибольший	<code>!\qquad</code> пробел наибольший

7.3. Символы, вводимые командами

Специальную группу символов можно ввести в текст с помощью команд. К этой группе относятся символы, которые при вводе “как есть” интерпретируются как управляющие, ряд знаков препинания, а также некоторые хитрые символы, буквы различных алфавитов сложного написания и символы со значками.

- Управляющие символы

`$` `#` `%` `&` `-` `^` `{` `}` `\`
вводятся “как есть” командами

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

- Двойные кавычки “...” вводятся как два символа ‘‘...’’, даже если клавиатура, с которой делается ввод, имеет соответствующие клавиши.
- Два типа черточек: обозначение числовых пределов и тире — вводятся как два или три символа -:
`--` `---`

- Многоточие ... вводится посредством команды `\dots`.

- Хитрые символы

`†` `‡` `©` `¶` `§` `ı` `ı`
вводятся командами

`\dag` `\ddag` `\copyright` `\P` `\S` `!‘` `?‘`

- Буквы различных алфавитов

`å` `Å` `æ` `Æ` `ı` `L` `ø` `Ø` `œ` `Œ` `ß`
вводятся командами:

`\aa` `\AA` `\ae` `\AE` `\l` `\L` `\o` `\O` `\oe` `\OE` `\ss`

- Символы с надстрочными диакритическими знаками

ò ó ô ö õ ô ò ò ò ò

вводятся так:

\‘{o} \’{o} \^{o} \^{o} \~{o} \={o} \.{o} \u{o} \v{o} \H{o}

- Подстрочные и связывающие диакритические знаки

q q q oo

вводятся так:

\c{o} \d{o} \b{o} \t{oo}

- Знак номера № вводится командой:

\numbersign

при задании корректирующего стиля `russian`.

Набор специальных символов, используемых в математической моде, рассмотрен в разделе “Работа с математическими выражениями”.

7.4. Команды выбора шрифтов

Нормальный шрифт документа, предназначенный для вывода основной массы текста, выбирается по умолчанию или указанием определенного файла корректирующего стиля: `10pt`, `11pt` или `12pt` (по умолчанию выбирается стиль `10pt`). Это всегда прямой шрифт. Размер нормального шрифта документа является важнейшей характеристикой, которая определяет выбор шрифтов для заголовков рубрик различного уровня, сносок и других стандартных элементов публикации.

Однако при работе с текстом для выделения тех или иных его фрагментов необходимы команды оперативной смены шрифтов.

Команды описания шрифтов делятся на две группы, декларирующие типы шрифтов и их размеры.

7.4.1. Задание типа шрифта

В приведенных здесь командах определения типа шрифта названия типов напечатаны шрифтом, вызываемым соответствующей командой:

\rm	Прямой шрифт	(Roman type style)
\bf	Жирный шрифт	(Bold type style)
\it	<i>Курсив</i>	<i>(Italic style style)</i>
\sl	<i>Наклонный шрифт</i>	<i>(Slanted type style)</i>
\sf	Рубленный шрифт	(San Serif type style)
\sc	КАПИТЕЛЬ	(SMALL CAPS TYPE STYLE)
\tt	Равноширинный шрифт	(Typewriter type style)

7.4.2. Задание размера шрифта

Команды определения размера шрифта и их значения в зависимости от выбора нормального шрифта представлены в таблице 2. Единицы измерения — типографские пункты.

Таблица 2. Стандартные размеры шрифтов в L^AT_EX’e.

Команда L ^A T _E X’a	Нормальный размер шрифта			
	10pt	11pt	12pt	14pt
<code>\tiny</code>	5	6	6	7
<code>\scriptsize</code>	7	8	8	9
<code>\footnotesize</code>	8	9	10	11
<code>\small</code>	9	10	11	12
<code>\normalsize</code>	10	11	12	14
<code>\large</code>	12	12	14	17
<code>\Large</code>	14	14	17	20
<code>\LARGE</code>	17	17	20	25
<code>\huge</code>	20	20	25	—
<code>\Huge</code>	25	25	—	—

Нормальный размер шрифта 14pt не является стандартным. Задание краткой формы (14pt) соответствующего файла в списке корректирующих стилей команды `\documentstyle` при стандартных базовых стилях недопустимо, следует указывать полное название файла, `art14`.

Будучи введенной в текст, команда задания размера или типа шрифта будет действовать вплоть до конца документа или до выдачи новой аналогичной команды. Для набора выбранным шрифтом фрагмента текста следует поместить в общие группирующие скобки и команду выбора шрифта, и сам фрагмент.

Будучи заданной в пределах командных скобок, любая команда выбора шрифта действует только в этих пределах.

Шелест, шорох, шопот, стон — смех, плач, пенье, крик, вопль, рев!

Размер нормального шрифта — 12pt.

```
{\tiny Шелест}, {\scriptsize шорох}, {\footnotesize шопот}, {\small стон} ---  
смех, {\large плач}, {\Large пенье}, {\LARGE крик}, {\huge вопль}, {\Huge рев!}  
\par \medskip Размер нормального шрифта --- 12pt.
```

Существуют также команды, выбирающие шрифт не по его признакам, а по названию его в библиотеке шрифтов. Существуют команды выбора отдельного символа шрифта по его номеру в таблице шрифта. Эти команды рассмотрены в разделе “Шрифты” нашего руководства, после описания коллекции шрифтов.

7.5. Команды рубрикации

Команды рубрикации предоставляют возможность разбивать текст на разделы и подразделы различных уровней, именно:

<code>\part</code>	—	часть	<code>\subsubsection</code>	—	подподраздел
<code>\chapter</code>	—	глава	<code>\paragraph</code>	—	параграф
<code>\section</code>	—	раздел	<code>\subparagraph</code>	—	подпараграф
<code>\subsection</code>	—	подраздел	<code>\appendix</code>	—	приложение

Команды рубрикации (кроме `\appendix`, формат и действие которой зависит от *базового стиля* документа) имеют одинаковый формат с одним обязательным параметром, который есть заголовок рубрики. Исключением является команда `\part`, заголовок в которой не обязателен.

Образец формата:

```
\section[<заголовок для оглавления>]{<заголовок>}
```

Необязательный параметр используется редко. В нем указывается краткий заголовок, который будет использован в оглавлении и колонтитуле — «бегущем» заголовке на верху каждой страницы. Если этот параметр опущен, то обязательный аргумент команды рубрикации будет выдан и в тексте, и в оглавлении, и, если этого требует формат страницы, как колонтитул.

L^AT_EX автоматически нумерует разделы и подразделы. Для статьи номера рубрик младше раздела включают номера старших рубрик вплоть до раздела, для доклада и книги старшей рубрикой, номер которой будет входить в младшие рубрики, является глава.

Если необходимо иметь заголовок без номера, следует воспользоваться *-формой команды рубрикации:

```
\chapter*[<заголовок для оглавления>]{<заголовок>}
```

Размеры шрифтов для заголовков рубрик, наличие бегущего заголовка или помещение в оглавление рубрик, заданных в *-форме, определяются в файле описания стиля. Стандартные базовые стили не выводят бегущий заголовок и автоматически вносят в оглавление только нумеруемые рубрики.

7.6. Построение абзаца

Абзацы отделяются друг от друга пустой строкой либо командой начала абзаца. Две или три пустые строки или несколько команд начала абзаца подряд не влияют на работу L^AT_EX'a. Величина абзацного отступа определена в соответствующем файле описания стиля.

▷ `\par`
Эквивалент пустой строки, разделяющей абзацы.

▷ `\indent`
Образует горизонтальный промежуток, равный ширине абзацного отступа. Эта команда может добавить абзацный отступ там, где по умолчанию он подавляется.

▷ `\noindent`

Подавляет абзацный отступ.

Абзацы отделяются друг от друга пустой строкой. Две или три пустые строки подряд не влияют на работу $\text{\LaTeX}$ 'а.

Величина абзацного отступа определена в файле описания стиля.

`\par` Абзацы отделяются друг от друга пустой строкой.

Две или три пустые строки подряд не влияют на работу $\text{\LaTeX}$ 'а.

`\noindent` Величина абзацного отступа определена в файле описания стиля.

7.7. Центрирование фрагмента текста

▷ `\begin{center}`

<текст на строке 1> `\`

.....

<текст на строке n>

`\end{center}`

Командные скобки для формирования абзаца из строк, центрированных относительно левого и правого полей страницы. Каждая строка должна заканчиваться командой `\`. Если текст для одной строки превышает ширину страницы, $\text{\LaTeX}$ расположит его на нескольких строках, каждая из которых будет центрирована на основании параметров, заданных по умолчанию.

▷ `\centering{<текст>}`

Команда центрирования текста без построения абзаца. Может быть использована внутри командных скобок, таких как `table`, `parbox` и прочих.

О ЦЕНТРИРОВАНИИ

Командные скобки центрирования текста просты в использовании.

Если текст для одной строки превышает ширину страницы, $\text{\LaTeX}$ расположит его на нескольких строках, каждая из которых будет центрирована на основании параметров, заданных по умолчанию.

```
\begin{center}\fbox{\parbox{0.6\hsize}
{\centering{\small \bf О ЦЕНТРИРОВАНИИ}}}\smallskip
{\sl Командные скобки центрирования текста просты в использовании.} \smallskip
Если текст для одной строки превышает ширину страницы,
\LaTeX\ расположит его на нескольких строках, каждая из которых
будет центрирована на основании параметров, заданных по умолчанию.\end{center}
```

7.8. Сдвиг фрагмента текста влево или вправо

▷ `\begin{flushleft}`
 <текст на строке 1> \\

 <текст на строке n>
`\end{flushleft}`

Командные скобки для формирования абзаца, строки которого сдвинуты к левому полю страницы. Каждая строка должна заканчиваться командой `\\`.

▷ `\raggedright`
В пределах строки сдвигает текст влево. Команда не формирует новый абзац.

▷ `\begin{flushright}`
 <текст на строке 1> \\

 <текст на строке n>
`\end{flushright}`

Командные скобки для формирования абзаца, строки которого сдвинуты к правому полю. Каждая строка должна заканчиваться командой `\\`.

▷ `\raggedleft`
В пределах строки сдвигает текст вправо. Команда не начинает новый абзац.

Нехорошо душе без знания, и торопливый ногами оступится.
Кто приобретает разум, тот любит душу свою.

Восточная мудрость о пользе учения.

```
\begin{flushleft}
Нехорошо душе без знания, и торопливый ногами оступится. \\
Кто приобретает разум, тот любит душу свою.\end{flushleft}
{\par \raggedleft Восточная мудрость о пользе учения.\par}
```

Восточная мудрость о пользе учения:

Нехорошо душе без знания, и торопливый ногами оступится.
Кто приобретает разум, тот любит душу свою.

```
{\par \raggedright Восточная мудрость о пользе учения:\par}
\begin{flushright} Нехорошо душе без знания, и торопливый ногами
оступится. \\ Кто приобретает разум, тот любит душу свою. \end{flushright}
```

7.9. Ввод горизонтальных промежутков

Для ввода в строку горизонтальных промежутков произвольных размеров имеются следующие команды.

▷ `\hspace{<длина>}`

▷ `\hspace*{<длина>}`

Команда задания промежутка определенной длины. Параметр *<длина>* является обязательным и должен быть задан размерной величиной. Длина может быть отрицательной. L^AT_EX игнорирует горизонтальный промежуток, который попадает в начало или конец строки. Для вставки промежутка, который не удаляется никогда, следует воспользоваться *-формой команды `\hspace`.

▷ `\hskip<длина>`

Примитив T_EX'a для задания на строке промежутка определенной длины. Аргумент *<длина>* является обязательным, задается *без фигурных скобок* и должен быть задан размерной величиной. Длина может быть отрицательной. L^AT_EX игнорирует горизонтальный промежуток, который попадает в начало строки в середине абзаца или в конец любой строки абзаца.

▷ `\hfill`

Примитив T_EX'a для создания эластичного промежутка, заполненного пробелами.

Следует отметить, что правильно команды вставки горизонтальных промежутков работают тогда, когда строка выделена в отдельный абзац. В приведенных ниже примерах каждая строка образует абзац.

		Текст смещен вправо.
Текст	Номер разделен промежутками.	Описание процесса
<code>\par \hfill Текст смещен вправо.</code>		
<code>\par \hspace*{4cm}{\sl Номер \hfill Описание процесса}\hspace*{4cm}</code>		
<code>\par Текст \hskip 12pt разделен \hskip 12pt промежутками.</code>		

7.10. Ввод вертикальных промежутков

Типы вертикальных промежутков на странице аналогичны горизонтальным промежуткам на строке: вертикальные промежутки типа пробелов, равные пропуску строки или части ее, промежутки, длина которых задается пользователем, и вертикальные промежутки эластичной длины.

▷ `\smallskip`

Вертикальный промежуток, равный пропуску четверти строки.

- ▷ `\medskip`
Вертикальный промежуток, равный пропуску половины строки.
- ▷ `\bigskip`
Вертикальный промежуток, равный пропуску строки.
- ▷ `\vspace{<длина>}`
- ▷ `\vspace*{<длина>}`
Команда задания промежутка определенной длины. Параметр `<длина>` обязателен и должен быть задан размерной величиной. `<Длина>` может быть отрицательной. `ЛATEX` игнорирует вертикальный промежуток, который попадает в начало или конец страницы. Для вставки промежутка, который не удаляется никогда, следует воспользоваться `*`-формой команды `\vspace`.
- ▷ `\vskip<длина>`
Примитив `TEX`'а для задания вертикального промежутка определенной длины. Аргумент `<длина>` является обязательным, задается *без фигурных скобок* и должен быть задан размерной величиной. Длина может быть отрицательной. `ЛATEX` игнорирует вертикальный промежуток, который попадает в начало или конец страницы.
- ▷ `\vfill`
Примитив `TEX`'а для создания эластичного промежутка, который может растягиваться или сжиматься по вертикали.
- ▷ `\addvspace{<длина>}`
Команда задания дополнительного вертикального промежутка. Может быть выдана только после пустой строки или команды `\par`. Параметр `<длина>` обязателен и должен быть размерной величиной. `<Длина>` может быть отрицательной. Команда используется, например, для изменения размера промежутка над или под фрагментом текста, помещенного в командные скобки, если они формируют абзац. При выдаче нескольких команд `\addvspace` подряд сработает одна, с максимальным значением параметра `<длина>`.

_____1	-----1
*_____2	\begin{flushleft}-----2
*_____3	*-----3
*_____4	*-----3
*_____5	\smallskip-----4
*_____6	*-----4
	\medskip-----5
	*-----5
	\bigskip-----6
_____7	*-----6
	\end{flushleft} \addvspace{15pt}\par-----7
	-----7
_____8	\par\vspace{1cm}-----8
	\hspace*{2cm}-----8

7.11. Цитирование

▷ `\begin{quotation}`
 <текст>
`\end{quotation}`

Введение в текст короткой цитаты. Поля командных скобок `quotation` имеют особый отступ слева и справа. Текст выравнивается по обоим полям.

▷ `\begin{quote}`
 <текст>
`\end{quote}`

Введение в текст цитаты. Поля командных скобок `quote` имеют особый отступ слева и справа. Текст выравнивается по обоим полям.

▷ `\begin{verse}`
 <строка 1> \\

 <строка n>
`\end{verse}`

Введение в текст стихотворной цитаты.

Сильное впечатление произвели на слушателей рассуждения Мертона, весьма мрачно настроенного современного философа:

“Ужас нашего времени составляет легкость, с которой всякие теории могут быть применены на практике. Чем идеальнее, чем совершеннее теория, тем кошмарнее выглядит ее реализация...”

Свобода завязана с несовершенством, ограничения, неувязки и ошибки не только неизбежны, но приветствуемы.”

Стихи удобно вводить командными скобками `verse`:

От луны лучи протягиваются,
К сердцу иглами притрагиваются.

В.Брюсов

Сильное впечатление произвели на слушателей рассуждения Мертона, весьма мрачно настроенного современного философа:

```
\begin{quotation}
\it ‘‘Ужас нашего времени составляет легкость, с которой всякие теории могут
быть применены на практике. Чем идеальнее, чем совершеннее теория, тем кошмарнее
выглядит ее реализация\ldots\par Свобода завязана с несовершенством, ограничения,
неувязки и ошибки не только неизбежны, но приветствуемы.’’
\end{quotation}
```

```

\bigskip
  Стихи удобно вводить командными скобками {\tt verse}:
      \begin{verse}
        От луны лучи протягиваются,    \\
        К сердцу иглами притрагиваются.\\
        \raggedleft {В.Брюсов}\\
      \end{verse}

```

7.12. Буквальная передача текста

▷ `\begin{verbatim}`
 <текст>
`\end{verbatim}`

Командные скобки создают абзац, выглядящий точно так, как он набран в исходном файле. Все символы, заключенные в эти командные скобки (в том числе пробелы) воспроизводятся в точности машинописным (равноширинным) шрифтом.

▷ `\begin{verbatim*}`
 <текст>
`\end{verbatim*}`

Если командные скобки `verbatim` заданы в форме со звездочкой, то пробелы будут напечатаны символом `␣`.

▷ `\verb|<текст>|`
 ▷ `\verb*|<текст>|`

Эти команды соответствуют командным скобкам `verbatim` и `verbatim*`. Печатают текст, заключенный между ограничителями, точно так, как он набран в исходном файле. В качестве ограничителей можно использовать любой символ, например `|`, так как он редко употребляется в обычных текстах. В отличие от командных скобок `verbatim` команда `\verb` не создает абзаца.

Команда `\hspace{3cm}` вводит горизонтальный промежуток.

Команда `\hspace{3cm}` вводит вертикальный промежуток.

Команда `\hspace` полезна.

Команда `\hspace` полезна.

```

\begin{verbatim}
Команда \hspace{3cm} вводит горизонтальный промежуток.
Команда \hspace{3cm} вводит вертикальный промежуток.
\end{verbatim}

```

```
\begin{verbatim}
Команда \hspace{3cm} вводит горизонтальный промежуток.
Команда \vspace{3cm} вводит вертикальный промежуток.
\end{verbatim}
```

```
\noindent \verb|Команда \hspace | полезна.
```

```
\noindent \verb*|Команда \vspace | полезна.
```

7.13. Составление списков

В L^AT_EX'e имеются несколько командных скобок для составления списков с перечнем пунктов. Каждый пункт в таких скобках начинается с команды `\item`. В пределах командных скобок должен быть по крайней мере один пункт.

По умолчанию принято, что командные скобки составления списков могут быть вложены одна в другую глубиной не более 4.

```
▷ \begin{description}
  \item [ключ 1] <Первый пункт>
  .....
  \item [ключ n] <Последний пункт>
\end{description}
```

Формирует список, каждый пункт выдается с новой строки с отступом больше абзацного. В квадратных скобках можно указать ключевое слово пункта, которое будет выделено жирным шрифтом.

```
▷ \begin{enumerate}
  \item <Первый пункт>
  .....
  \item <Последний пункт>
\end{enumerate}
```

Формирует список с пронумерованными пунктами. Пункты списка первого уровня отмечаются цифрами, второго — буквами, третьего — малыми латинскими цифрами, четвертого — заглавными буквами.

```
▷ \begin{itemize}
  \item [метка 1]<Первый пункт>
  .....
  \item [метка n] <Последний пункт>
\end{itemize}
```

Формирует список, каждый пункт которого начинается с маркера. В квадратных скобках можно указать метку, которой будет маркирован данный пункт. Принятый по умолчанию маркер первого уровня — символ •.

1. Что такое руководство по программному обеспечению.	<code>\begin{enumerate}</code>
• Кто пишет руководства?	<code>\item Что такое руководство по программному обеспечению.</code>
• Как оно должно выглядеть?	<code>\begin{itemize}</code>
Внешний вид важен.	<code>\item Кто пишет руководства?</code>
Содержание еще более важно.	<code>\item Как оно должно выглядеть?</code>
2. Важность организации.	<code>\begin{description}</code>
• Организация текста.	<code>\item [Внешний вид] важен.</code>
а) Организация справочной информации.	<code>\item [Содержание] еще более важно.</code>
– Наличие каталогов.	<code>\end{description}</code>
– Индексный указатель.	<code>\end{itemize}</code>
б) Организация обучающего материала.	<code>\item Важность организации.</code>
	<code>\begin{itemize}</code>
	<code>\item Организация текста.</code>
	<code>\begin{enumerate}</code>
	<code>\item Организация справочной информации.</code>
	<code>\begin{itemize}</code>
	<code>\item Наличие каталогов.</code>
	<code>\item Индексный указатель.</code>
	<code>\end{itemize}</code>
	<code>\item Организация обучающего материала.</code>
	<code>\end{enumerate}</code>
	<code>\end{itemize}</code>
	<code>\end{enumerate}</code>

8. Горизонтальные и вертикальные линии

Л^AT_EX рассматривает любую линию как полосу определенной длины и толщины, положение которой определено относительно базовой линии строки. Горизонтальные линии — это линии с большой длиной и малой толщиной, вертикальные линии имеют малую длину и большую толщину. Ввести в документ горизонтальную или вертикальную линию можно следующей командой.

▷ `\rule [<положение>] {<длина>} {<толщина>}`

Обязательные параметры команды задают соответственно длину и толщину линии. Необязательный первый параметр определяет, как высоко линия поднята (или опущена, если он отрицательный) относительно базовой линии строки.

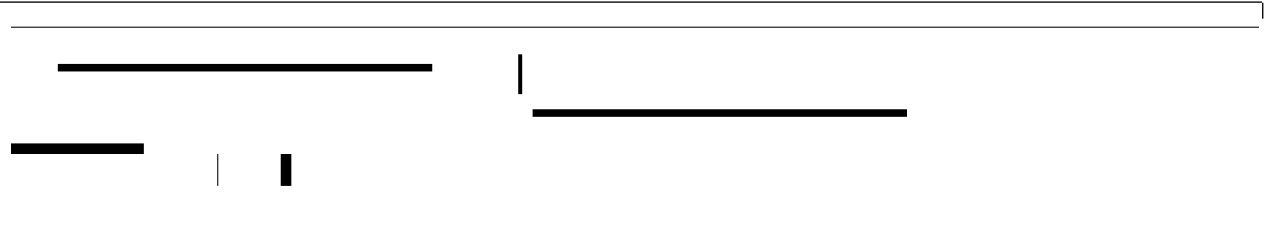
T_EX предлагает для проведения линий две команды, параметры которых: `width`, `height`, `depth` — не являются обязательными и имеют смысл длины, толщины и положения относительно базовой линии строки.

▷ `\hrule width<величина> height<величина> depth<величина>`

Команда T_EX'a для задания горизонтальной линии. По умолчанию будет установлена ширина в полный размер строки, высота в 0.4 pt и глубина 0 pt. Команда приводит к окончанию абзаца.

▷ `\vrule width<величина> height<величина> depth<величина>`

Команда T_EX'a для задания вертикальной линии. По умолчанию будет установлена ширина в 0.4 pt, высота в текущий размер высоты строки или бокса и глубина в 0 pt. В отличие от предыдущей, данная команда не приводит к окончанию абзаца.



```
\hrule \vspace {10pt}
\rule[+3mm]{0.3\hsize}{1mm}\hspace{1cm} \rule{1.5pt}{15pt}
\rule[-3mm]{0.4\hsize}{1mm}\hspace{1cm} \par \vspace {10pt}
\hrule width 50pt height 2pt depth 2pt
\hspace{60pt}\vrule \qqquad \vrule width 4pt height 6pt depth 6pt
```

9. Плавающие элементы: рисунки и таблицы

Рисунки и таблицы (если последние занимают не более одной страницы) не могут быть разорваны между страницами. Поэтому $\text{\LaTeX}$ должен иметь возможность двигать эти элементы, если они заданы так, что помещение их в точке выдачи команды невозможно — нехватает места.

Для правильной верстки текста с рисунками и таблицами в $\text{\LaTeX}$ ’е имеются командные скобки, объявляющие содержимое их плавающим элементом типа рисунка или типа таблицы. Местоположение таких элементов в сверстанном документе зависит от их размера и места выдачи соответствующей команды в исходном файле.

$\text{\LaTeX}$ приписывает плавающим элементам порядковые номера, отдельно для таблиц и для рисунков. При выдаче в пределах командных скобок плавающего элемента команды `\caption`, определяющей заголовок данного элемента, его номер вводится в заголовок автоматически.

В списки рисунков и таблиц $\text{\LaTeX}$ включает *только* те, которые определены как плавающие элементы.

Так как командные скобки плавающих элементов имеют номера, $\text{\LaTeX}$ позволяет вводить на них в тексте ссылки.

Командные скобки для вставки рисунков как плавающих элементов:

```
▷ \begin{figure}[<место>]
    <рисунок>
    \caption{<название рисунка>}
\end{figure}
```

Командные скобки для вставки таблиц как плавающих элементов:

```
▷ \begin{table}[<положение>]
    <тело таблицы>
    \caption{<название>}
\end{table}
```

Единственным параметром командных скобок плавающего элемента является указание желаемого места появления этого элемента на странице. Этот параметр не является обязательным и может быть задан посредством нижеприведенных символов или их комбинации:

- | | |
|--------------|--|
| h (здесь) | позиция в тексте, где появляется командная скобка; |
| t (вверху) | в начале следующей страницы; |
| b (внизу) | в конце следующей страницы; |
| p (страница) | на отдельной странице, содержащей только плавающие элементы. |

По умолчанию стандартные стили используют спецификатор `tbp`.

Команда `\caption` позволяет поименовать рисунок или таблицу в принятом для выбранного базового стиля формате. Она может быть задана и над элементом.

Если вы хотите сослаться на таблицу или рисунок, в соответствующие командные скобки *после* команды `\caption` поместите команду `\label` с заданием метки. Формат команды `\label` описан в разделе 12 о перекрестных ссылках.

L^AT_EX вставляет плавающий элемент, используя ряд *параметров стиля*, которые могут препятствовать его вводу в желаемое место, даже если визуально очевидно, что этого места достаточно. Если вы решительно хотите иметь рисунок или таблицу в определенном месте документа, в список корректирующих стилей команды `\documentstyle` следует ввести стиль `here` и указать в команде `\figure` в качестве параметра положения H (прописное!).

Например, ввод таблицы 2 сделан следующими командами (корректирующий стиль `here` задан в команде `documentstyle` исходного файла нашей книги):

```
\begin{table} [H]
\begin{center}
<тело таблицы>
\end{center}
\caption{Стандартные размеры шрифтов в \LaTeX'e.} \label{t-one}
\end{table}
```

10. Оглавление, списки рисунков и таблиц

Базовые стили `article`, `report`, `book` вводят в оглавление только нумерованные рубрики (те, что заданы простым, а не `*`-форматом команд рубрикации), а в списки рисунков и таблиц — только те, что сформированы в пределах командных скобок `figure` и `table`.

Команда ввода оглавления имеет вид:

▷ `\tableofcontents`

Для построения оглавления $\text{\LaTeX}$ ’у необходимо обработать текст дважды. В первый раз во вспомогательный файл, имеющий то же имя, что и исходный файл, но расширение `toc`, будут занесены заголовки рубрик. При втором проходе $\text{\LaTeX}$ ’а по данным этого файла будет сформировано оглавление.

Список рисунков и список таблиц создаются следующими командами:

▷ `\listoffigures`

Создает список рисунков, описанных в командных скобках `figure`.

▷ `\listoftables`

Создает список таблиц, описанных в командных скобках `table`.

При составлении списка рисунков или таблиц $\text{\LaTeX}$ ’у обрабатывает текст дважды. В первый раз необходимые данные будут занесены во вспомогательные файлы, имеющие то же имя, что и исходный файл, но различные расширения: в файл `lof` будут занесены подписи к рисункам, в файл `lot` — названия таблиц. При втором проходе $\text{\LaTeX}$ ’а по данным этих файлов будут сформированы надежащие списки.

Если оглавление, список таблиц, список рисунков должны начинаться с новой страницы, перед соответствующей командой вставьте команду `\newpage`.

Может случиться, что в оглавление или в список рисунков или таблиц надо вставить пункты, не помещаемые туда автоматически. Для этой цели существуют следующие команды.

▷ `\addcontentsline{<файл>}{<тип секции>}{<вход>}`

Добавляет вход к указанному списку. Параметр `<файл>` определяет расширение файла, в который будет добавляться информация: список оглавлений (`toc`), список рисунков (`lof`), список таблиц (`lot`). `<Тип секции>` управляет формированием входа и должен быть для файла `toc` типом рубрики (например, `part` или `section`), для файла `lof` — `figure`, для файла `lot` — `table`. Параметр `<вход>` задает текст данного входа.

▷ `\addtocontents{<файл>}{<текст>}`

Добавляет `<текст>` к файлу, который строит список оглавлений или список рисунков или таблиц. Параметр `<файл>` определяет расширение файла, в который будет добавляться информация: список оглавлений (`toc`), список рисунков (`lof`), список таблиц (`lot`).

▷ `\nofiles`

Команда запрещения записи в рабочие файлы `aux`, `toc`, `lof` `lot`. Существенно ускоряет работу при отладочных прогонах $\text{\LaTeX}$ 'а.

Пусть стиль `report` устраивает вас по всем параметрам, кроме одного: название рубрики «часть» печатается на отдельной страницы, тогда как надо иметь ее над текстом. Следует сформировать заголовок и сопроводить его надлежащей командой `\addcontentsline`. Заголовок будет вынесен в оглавление в виде, принятом для рубрики «часть».

Если заголовок, заданный *-формой команды рубрикации, нужен в оглавлении, поместит его туда опять-таки команда `\addcontentsline`. Заголовок появится в оглавлении без номера, в виде, принятом для данной рубрики.

```
\par{\centering{Часть I. ВВЕДЕНИЕ В ТЕОРИЮ МНОЖЕСТВ}}\par
\addcontentsline{toc}{part}{ВВЕДЕНИЕ В ТЕОРИЮ МНОЖЕСТВ}
.....
\section*{Некоторые ограничения}
\addcontentsline{toc}{section}{Некоторые ограничения}
.....
```

11. Ввод библиографии

Процесс ввода библиографических данных, как правило, включает в себя не только формирование списка литературы, но и расстановку соответствующих ссылок. Для автоматизации этих процедур $\text{\LaTeX}$ предоставляет следующие средства.

11.1. Построение списка литературы в тексте документа

```
▷ \begin{thebibliography}{<метка наибольшей ширины>}
  \bibitem[<метка 1>]{<ключ_ссылки 1>}
    <описание библиографического входа 1>
    .....
  \bibitem[<метка nn>]{<ключ_ссылки nn>}
    <описание библиографического входа 2>
\end{thebibliography}
```

Командные скобки создания списка литературы. Обязательный параметр указывает, сколько места надо отводить под максимальный номер или метку элементов списка.

```
▷ \bibitem[<метка>]{<ключ_ссылки>}
```

Задаёт входы с описанием литературных источников в пределах командных скобок `thebibliography`. Обязательный параметр `<ключ_ссылки>` есть ключ, используемый в команде ссылки на данную работу. Таким ключом может быть любая последовательность из букв, чисел и знаков пунктуации, не содержащая запятую.

Необязательный параметр `<метка>` определяет метку, которая будет присвоена входу в списке литературы. Если эта метка отсутствует, входу будет присвоен текущий порядковый номер, определенный счетчиком перечисления.

Команда `\bibitem` записывает параметры входа в файл с расширением `aux`. При повторном прогоне текста через $\text{\LaTeX}$ этот файл читается командой `\begin{document}`, и для каждого элемента списка литературы по команде `\cite`, ссылающейся на параметр `<ключ_ссылки>`, будет подставлена `<метка>`, определяющая его место в списке литературы.

Список литературы

- [1] Donald E. Knuth: *The $\text{\TeX}$ Book*. Addison-Wesley PC. 1984.
- [2] Donald E. Knuth: *The METAFONTbook*. Addison-Wesley PC. 1984.
- [3] И.А.Грицаенко, В.К.Мальшев и др. — Препринт ИФВЭ 91-55, Протвино, 1991.

```
\begin{thebibliography}{99}
\bibitem{knuth-84}
Donald E. Knuth: {\it The \TeX Book}.
Addison-Wesley PC. 1984.
\bibitem{knuth-meta}
Donald E. Knuth: {\it The METAFONTbook}.
Addison-Wesley PC. 1984.
\bibitem{91-55}
И.А.Грицаенко, В.К.Мальшев и др.
--- Препринт ИФВЭ 91-55, Протвино, 1991.
\end{thebibliography}
```

11.2. Расстановка ссылок на литературу

▷ `\cite[<текст>]{<список меток>}`

Команда ссылки на один или несколько пунктов списка литературы. Обязательный параметр должен содержать одну или несколько меток, разделенных запятыми или тире. При наличии файла `aux`, связывающего эти метки с метками или номерами в списке литературы, в текст будут вставлены надлежащие ссылки на этот список. Как упоминалось ранее, процедура расстановки ссылок требует двух прогонов $\text{\LaTeX}$ 'а. Необязательный параметр `<текст>` будет представлен как комментарий к ссылке.

12. Перекрестные ссылки

В документах большого объема со значительным количеством рубрик и подразбук, таблиц, рисунков, уравнений и т.д., перекрестные ссылки на те или иные фрагменты текста являются важным и органичным элементом.

▷ `\label{<ключ>}`

Команда задания метки определенного элемента текста.

Будучи задана в пределах нумеруемых командных скобок, команда `\label` присваивает метке номер этих скобок. К нумеруемым командным скобкам $\text{\LaTeX}$ в текстовой моде относит скобки ввода плавающих элементов, а в математической моде — скобки ввода уравнений и теорем. Для каждого типа таких скобок имеется счетчик, позволяющий задавать соответствующей скобке порядковый номер.

Если команда появляется вне командных скобок, то метке `<ключ>` присваивается номер текущей рубрики.

Метка может состоять из последовательности букв, цифр или знаков препинания. Прописные и строчные буквы различаются.

▷ `\ref{<ключ>}`

Приводит к печати номера раздела, уравнения и т.п., отмеченного командой `\label` с той же меткой `<ключ>`.

▷ `\pageref{<ключ>}`

Приводит к печати номера страницы, в пределах которой выдана команда `\label` с той же меткой `<ключ>`.

Заголовок данной рубрики в исходном файле выглядит так:

```
\subsection{Перекрестные ссылки}\label{ref}
```

Команды задания перекрестных ссылок описаны на странице 39, в разделе 12.

Распознаваемые $\text{\LaTeX}$ 'ом единицы измерения жесткой длины и соотношения между ними приведены в таблице 1 на странице 9.

Команды задания перекрестных ссылок описаны на странице `\pageref{ref}`, в разделе `\ref{ref}`.

`\par\bigskip`

Распознаваемые $\text{\LaTeX}$ 'ом единицы измерения жесткой длины и соотношения между ними приведены в таблице `\ref{t-two}` на странице `\pageref{t-two}`.

13. Построение мини-страницы

В пределах страницы L^AT_EX позволяет построение мини-страниц — страниц малого размера.

Мини-страницы аналогичны полной странице и могут состоять из абзацев и содержать большинство командных скобок. Они занимают лишь определенную пользователем часть полной страницы. На одной странице может быть несколько мини-страниц.

Командные скобки построения мини-страницы не создают отдельного абзаца. Мини-страница создается непосредственно в точке выдачи команды начала соответствующих командных скобок и может быть создана, например, в середине строки.

▷ `\begin{minipage}[<позиция>] {<ширина>}`
`<текст>`
`\end{minipage}`

Командные скобки построения мини-страницы имеют один обязательный параметр, указывающий ширину. Высота мини-страницы определяется размером помещаемого на ней текста. Необязательный параметр `<позиция>` допускает следующие значения:

- c мини-страница будет центрирована относительно текстовой строки;
- t верхняя строка мини-страницы выравнивается по текстовой строке;
- b последняя строка мини-страницы выравнивается по текстовой строке.

Две мини-страницы центрированы относительно текстовой строки. Ширина каждой страницы задана в долях ширины полной страницы. Размер боксов на мини-страницах задан в долях ширины этих страниц.

Моральная устойчивость определяет непреклонность и силу воли персонажа, его отвагу и способность сопротивления чужим воздействиям.

Глава 1.....	1
*** 1.1 Основные начала	2
1.2 Технология	5
Глава 2.....	8

```
\begin{minipage}{0.65\hsize}
  \begin{center}
    \fbox{\parbox{0.95\hsize}{\bf
      Моральная устойчивость определяет непреклонность и силу воли персонажа,
      его отвагу и способность сопротивления чужим воздействиям.}}
  \end{center}
\end{minipage} **** \begin{minipage}{0.4\hsize}
\ vbox{ \ hbox to 0.8\hsize { Глава 1.\hrulefill\ 1}
        \ hbox to 0.8\hsize { 1.1 Основные начала\dotfill\ 2}
        \ hbox to 0.8\hsize { 1.2 Технология\dotfill\ 5}
        \ hbox to 0.8\hsize { Глава 2.\hrulefill\ 8}}
\end{minipage}
```

14. Сноски и заметки на полях

Сноски в подавляющем большинстве случаев — на страницах и мини-страницах, в пределах командных скобок — создаются командой `\footnote`.

Только во фрагментах текста, помещенных внутри бокса, сноски формируются двумя командами: `\footnotemark` и `\footnotetext` — порядок которых существенен. Первая из них задается внутри бокса для указания места метки, а вторая, с собственно текстом сноски, должна быть задана вне бокса.

▷ `\footnote[<номер>] {<текст>}`

Формирует сноску¹. Необязательный аргумент `<номер>` используется, чтобы изменить номер, задаваемый по умолчанию. Команда не может использоваться внутри бокса, за исключением бокса, созданного скобками построения мини-страницы.

Описание команды `\footnote` в исходном файле выглядит так:

Формирует сноску `\footnote{Исходный текст этой строки со сноской дан в примере.}`.

Сноски на мини-страницах:

Сноски в пределах таблицы		
Элемент 11	Элемент 12 ^a	Элемент 13
Элемент 21	Элемент 22	Элемент 23
Элемент 31	Элемент 32 ^b	Элемент 33

Сноски на мини-страницах имеют свой счетчик и нумеруются лишь в пределах страницы.^c

^aЭлемент 12 отмечен сноской

^bЭлемент 32 отмечен сноской

^cПример это демонстрирует.

Внутри командных скобок центрирования текста^a сноски допустимы.

^aЭто утверждение относится к большинству командных скобок

```
\begin{center}
\sl Сноски на мини-страницах:
\end{center}

\begin{minipage}{0.5\hsize}
\par\bigskip
\begin{tabular}{|c|c|c|}
\hline
\multicolumn{3}{|c|}{Сноски в пределах таблицы} \\
\hline
Элемент 11 & Элемент 12\footnote{Элемент 12 отмечен сноской} & Элемент 13 \\
Элемент 21 & Элемент 22 & Элемент 23
\end{tabular}
\end{minipage}
```

¹Исходный текст этой строки со сноской дан в примере.

```

Элемент 31 & Элемент 32\footnote{Элемент 32 отмечен сноской} & Элемент 33 \\
\hline
\end{tabular}

```

```
\medskip
```

Сноски на мини-страницах имеют свой счетчик и нумеруются лишь в пределах страницы\footnote{Пример это демонстрирует.}.

```
\end{minipage}\hfill\begin{minipage}[b]{0.4\hsize}
```

```
\begin{center}
```

Внутри командных скобок центрирования текста\footnote{Это утверждение относится к большинству командных скобок}~ сноски допустимы.

```
\end{center}
```

```
\end{minipage}
```

▷ \footnotemark[<номер>]

Устанавливает номер и формирует маркер сноски, текст которой содержится в выданной далее команде \footnotetext. Необязательный аргумент <номер> используется для изменения номера, данного сноске по умолчанию, и должен быть выражен числом.

▷ \footnotetext[<номер>] {<текст>}

Формирует сноску, маркер которой уже задан командой \footnotemark. Необязательный аргумент <номер> используется для изменения номера, даваемого по умолчанию.

Сноска к фрагменту, помещенному командой \fbox в бкс² с рамкой³ формируется командами:

```
\fbox{ бкс\footnotemark~ } \footnotetext{Это сноска в первом бксе.}
```

```
\fbox{ рамкой\footnotemark~ } \footnotetext{Это сноска во втором бксе.}
```

Для ввода заметок на полях существуют следующие команды:

▷ \marginpar[<левая заметка>] {<правая заметка>}

Создает заметки на полях. <Левая заметка> может быть опущена. Для двусторонней одноколоной печати <правая заметка> помещается на внешнем поле, <левая заметка> — на внутреннем.

▷ \reversemarginpar

Включает такую моду работы команды \marginpar, при которой правая и левая заметки меняются местами.

▷ \normalmarginpar

Отменяя действие команды \reversemarginpar, устанавливает нормальное расположение заметок.

²Это сноска в первом бксе.

³Это сноска во втором бксе.

Нижеследующий абзац дает пример, позволяющий судить о степени выделенности текста, набранного с удвоением промежутка между строками.

Разрядка строк такого рода придает тексту особый вес; с ее помощью удобно указывать, что *не надо* делать, особенно со ссылками на работы, где применялись подобные неудачные действия.

*Способ
далеко
не лучший!*

Нижеследующий абзац дает пример, позволяющий судить о степени выделенности текста, набранного с удвоением промежутка между строками. \marginpar{\small \it Способ далеко не лучший!}

```
\begin{center}
\parbox{0.8\textwidth}{\par\baselineskip=2\normalbaselineskip\par
Разрядка строк такого рода придает тексту особый вес; с ее помощью удобно
указывать, что {\it не надо} делать, особенно со ссылками на работы, где
применялись подобные неудачные действия.
\par\baselineskip = \normalbaselineskip \par}
\end{center}
```

15. Боксы

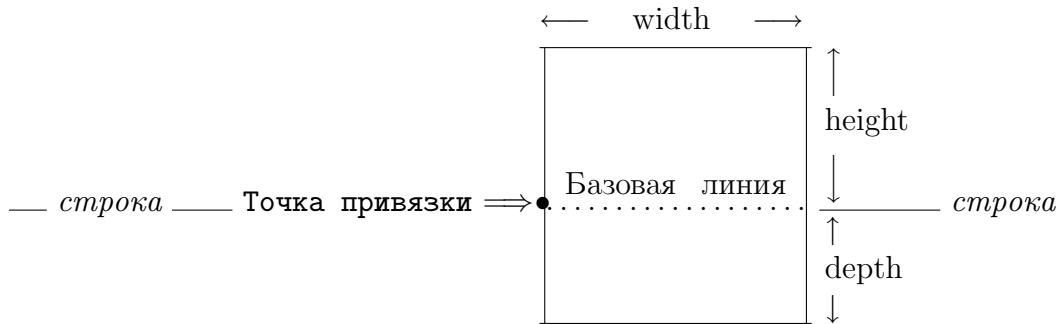
В данном разделе описаны команды, обрабатывающие фрагменты текста специальным образом. При выполнении этих команд $\text{\LaTeX}$ переходит в режим работы, называемый модой слева–направо, или LR модой.

В LR моде $\text{\LaTeX}$ считает, что обрабатывает последовательность слов и промежуточных, которые располагаются слева направо и не могут быть разбиты на строки. В этой моде $\text{\LaTeX}$ никогда не начинает новую строку. Даже если поместить сто слов внутри бокса, например, созданного командой `\mbox`, $\text{\LaTeX}$ будет продолжать компоновать их слева направо, а затем пожалуется, что полученный бокс такой большой, что не уместится на строке.

15.1. Что такое боксы

Основными понятиями, которыми $\text{\TeX}$ и $\text{\LaTeX}$ оперируют при построении строк, абзацев, страниц, являются *бокс* и *клей*. *Бокс* — это единица построения текста, некий неделимый объект; *клей* есть субстанция, заполняющая пространство между боксами в соответствии с надлежащими командами формирования текста и тем самым составляющая из боксов строки, из строк абзаца, из абзацев страницы.

Боксы представляют собой объекты прямоугольной формы, точка привязки которых расположена на пересечении базовой линии строки и левой стороны бокса. Относительно точки привязки бокс характеризуется тремя параметрами: высотой (*height*), шириной (*width*) и глубиной (*depth*).



Для $\text{\TeX}$ 'а боксами являются, например, символы любого шрифта (процессор работает только с метрическими файлами шрифтов, где для каждого символа указаны его ширина, высота, глубина и параметры, определяющие взаимные расстояния между символами, и не проверяет, существует ли на самом деле шрифт с заданной метрикой. С собственно шрифтовыми файлами работают драйверы).

15.2. Команды работы с боксами

В $\text{\LaTeX}$ 'е и $\text{\TeX}$ 'е существуют команды, явным образом помещающие в бокс отдельный элемент или фрагмент текста. Помещение фрагмента текста в бокс является отличным способом выделения этого фрагмента, так как позволяет не только изменять ширину его и местоположение на странице, но и помещать фрагмент в рамку.

—
Здесь приведены команды работы с боксами и команды, которые можно использовать только внутри боксов.

- ▷ `\mbox{<текст>}`
Создает бокс, ширина которого определяется величиной параметра `<текст>`.
- ▷ `\makebox [<ширина>] [<положение>] {<текст>}`
Создает бокс с указанным текстом. Ширина бокса определяется шириной текста или необязательным параметром `<ширина>`. Позиция текста в боксе определяется необязательным параметром `<положение>`: по центру (c), слева (l), справа (r). По умолчанию текст центрируется.
- ▷ `\fbox{<текст>}`
Строит и обводит сплошной рамкой бокс с указанным текстом. Ширина бокса определяется параметром `<текст>`.
- ▷ `\framebox [<ширина>] [<положение>] {<текст>}`
Строит и обводит сплошной рамкой бокс с указанным текстом. Ширина бокса определяется параметром `<текст>` или необязательным параметром `<ширина>`. Позиция текста в боксе определяется необязательным параметром `<положение>`: по центру (c), слева (l), справа (r). По умолчанию текст центрируется.
- ▷ `\parbox [<положение>] {<ширина>} {<текст>}`
Создает бокс, содержимое которого формируется в моде абзаца. Имеет два обязательных параметра, определяющих ширину бокса и текст помещаемого в него абзаца. По умолчанию L^AT_EX располагает этот бокс так, что его центр находится на базовой линии строки, в которой выдана команда. Необязательный параметр `<положение>` позволяет смещать бокс относительно базовой линии вверх (t) и вниз (b). Команда `\parbox` предназначена для небольших простых порций текста. В частности, текст не должен содержать никаких командных скобок.
- ▷ `\raisebox {<высота>} [<выступ_вверх>] [<выступ_вниз>] {<текст>}`
Используется для смещения текста относительно базовой линии строки. Первый обязательный аргумент указывает, как высоко будет поднят текст, заданный в последнем параметре (или опущен, если его величина отрицательна). Необязательные параметры указывают, насколько `<текст>` выступает над базовой линией строки и насколько простирается ниже базовой линии.

Следующие команды используются только внутри боксов.

- ▷ `\dotfill`
Создает внутри бокса промежуток, заполненный точками. Размер его определяется шириной бокса.
- ▷ `\hrulefill`
Создает внутри бокса промежуток, заполненный горизонтальной линией. Размер его определяется шириной бокса.

▷ `\downbracefill`

Создает внутри бокса горизонтальную фигурную скобку, обращенную вниз. Размер ее определяется шириной бокса.

▷ `\upbracefill`

Создает внутри бокса горизонтальную фигурную скобку, обращенную вверх. Размер ее определяется шириной бокса.

Защита диссертации состоится _____ 1994 г.

.....

$$S_{\text{равномерного_движения}} = V \cdot t \text{ см}$$

Строка

*Чисти зубы ве-
чером, а утром
полощи.*

строка

*Чисти зубы ве-
чером, а утром
полощи.*

строка

*Чисти зубы ве-
чером, а утром
полощи.*

```

\par Защита диссертации состоится \makebox[3.5cm]{\hrulefill}~1994~г.
\par \medskip \makebox[0.95\hsize]{\dotfill}
\begin{center}
\parbox{3cm}{\downbracefill \ \ 1 2 3 4 5 6 7 8 \ \ \upbracefill} \ \
{\Large $ S_{\text{равномерного\_движения}}=V \cdot t\mbox{\thinspace см} $}
\end{center}
\medskip
Строка \fbox{\parbox[t]{0.18\hsize}
{\it Чисти зубы вечером, а утром полощи.}}
строка \fbox{\parbox{0.18\hsize}
{\it Чисти зубы вечером, а утром полощи.}}
строка \fbox{\parbox[b]{0.18\hsize}
{\it Чисти зубы вечером, а утром полощи.}}

```

Л^AT_EX может оперировать командами построения боксов T_EX'а. Здесь мы приводим описание двух команд-примитивов, лежащих в основе многих макрокоманд построения боксов: `\hbox` и `\vbox`.

- `\hbox to <величина> {<текст>}`

Команда образует горизонтальный бокс для указанного в качестве обязательного параметра текста, который может содержать команды T_EX'а. (В частности, именно в пределах горизонтального бокса допустимо использование команд `\hrulefill` и `\dotfill`, заполняющих соответствующую часть бокса сплошной или пунктирной линией). Необязательный параметр задает длину горизонтального бокса, при его отсутствии размер бокса определяется длиной текста. Параметр `<величина>`

может быть выражен как в принятых в TeX'e единицах размерности, так и через команду, определенную ранее.

- `\vbox to <величина> {<текст>}`

Команда, позволяющая строить вертикальные боксы. Для этого содержимое фигурных скобок должно быть текстом, распределенным по горизонтальным боксам. Команда `\vbox` расположит эти боксы один под другим. Необязательный параметр задает высоту вертикального бокса, при его отсутствии размер бокса определяется конструкцией параметра `<текст>`.

Глава 1.....	1
1.1 Основные начала.....	2
1.2 Технология.....	5
Глава 2.....	8

Содержимое бокса 1

Содержимое бокса 2 Содержимое бокса 4

Содержимое бокса 3 Содержимое бокса 5 Содержимое бокса 6

```
\vbox{ \hbox to 0.8\hsize { Глава 1.\hrulefill 1}
      \hbox to 0.8\hsize { 1.1 Основные начала\dotfill 2}
      \hbox to 0.8\hsize { 1.2 Технология\dotfill 5}
      \hbox to 0.8\hsize { Глава 2.\hrulefill 8}}
\par\vbox{\hbox{\hbox{Содержимое бокса 1}\hbox{Содержимое бокса 2}
\hbox{Содержимое бокса 3}}\hspace 1cm \vbox{\hbox{Содержимое бокса 1}
\hbox{Содержимое бокса 2}}}
```

15.3. Хранение текста в боксе

Нижеследующие команды дают удобный способ хранения информации, расположение которой в тексте не определено заранее или которая воспроизводится несколько раз. Такую информацию можно поместить в бокс командой, выданной, например, в начале документа, а в самом тексте иметь одну или несколько команд ссылки на этот бокс.

- ▷ `\newsavebox{\<имя>}`

Объявляет новый бокс с заданным именем. Символ `\` перед именем бокса является обязательным.

- ▷ `\savebox{\<имя>} [<ширина>] [<положение>] {<текст>}`

Помещает текст в бокс с именем `\<имя>`, который должен быть создан командой `\newsavebox`. По умолчанию L^AT_EX располагает бокс так, что его центр находится на базовой линии строки текста. Необязательный параметр `<ширина>` задает горизонтальный

размер бокса. Необязательный параметр <положение> позволяет смещать бокс относительно базовой линии вверх (t) и вниз (b).

▷ `\usebox{\<имя>}`

Печатает текст, помещенный в бокс `\<имя>` командой `\savebox`.

Пусть требуется напечатать приглашения на вечер бывших одноклассников.

Дорогой друг! Ждем Вас в нашей родной школе 25 февраля 1995 года. Ваши одноклассники	Дорогой друг! Ждем Вас в нашей родной школе 25 февраля 1995 года. Ваши одноклассники	Дорогой друг! Ждем Вас в нашей родной школе 25 февраля 1995 года. Ваши одноклассники
--	--	--

```
\newsavebox{\party}
\savebox{\party}[5cm]{\fbox{\parbox{4.8cm}
{\centerline{Дорогой друг!}
Ждем Вас в нашей родной школе 25 февраля 1995\thinspace года.\
\hspace*{0.5cm} Ваши одноклассники} }}

\noindent\usebox{\party}\quad \usebox{\party}\quad \usebox{\party}
```

16. Таблицы

Для построения таблиц $\text{\LaTeX}$ имеет две командные скобки:

- **tabbing**, имитирующие составление таблицы на пишущей машинке;
- **tabular**, позволяющие строить таблицы более свободно, не привязываясь к шаблону.

16.1. Выравнивание текста по шаблону

```
▷ \begin{tabbing}
    <вход 1> \= <вход 2> \= . . . \= <вход n> \\ % Строка-шаблон
    <вход 11> \> <вход 12> \> . . . \=><вход 1n> \\
    .....
\end{tabbing}
```

С помощью командных скобок **tabbing** можно выровнять текст по столбцам. Для этого нужно расставить табуляторы в строке-шаблоне, задающей способ выравнивания по столбцам — подобно тому, как это делается на обычной печатной машинке. Строка-шаблон должна быть первой строкой после начала командных скобок. Далее, однако, можно менять способ выравнивания, вводя новую строку шаблона.

Скобки **tabbing** используют контрольные символы:

<code>\=</code>	устанавливает табуляторы;
<code>\></code>	приводит к печати последующего текста и продвижению к следующему табулятору справа;
<code>\\</code>	начинает новую строку;
<code>\kill</code>	позволяет не печатать текст текущей строки, учитывая содержащиеся в ней контрольные символы, которые будут влиять на последующие строки. С помощью этой команды можно, например, избежать печати строки-шаблона;
<code>\+</code>	передвигает левое поле всех последующих строк на один табулятор вправо, как бы вставляя в начало их контрольный символ <code>\></code> . Несколько команд <code>\+</code> дадут суммарный эффект;
<code>\<</code>	сдвигает табулятор на шаг влево. Используется для строк, для которых была выдана одна или несколько команд <code>\+</code> ;
<code>\-</code>	передвигает левое поле последующих строк на один табулятор влево, отменяя таким образом одну выданную ранее команду <code>\+</code> ; Несколько команд <code>\-</code> дадут суммарный эффект;
<code>\'</code>	передвигает весь заданный после нее текст влево до правого края предыдущего столбца или до левого поля страницы;

\‘ передвигает вправо вплоть до правого поля страницы последующий текст, который не должен содержать никаких контрольных символов кроме символа конца строки.

`\pushtabs` спасает текущее значение шаблона.

`\poptabs` восстанавливает значение шаблона, спасенное по команде `\pushtabs`.

Команды `\-`, `\=`, `\'`, `\'` в пределах командных скобок `\tabbing` переопределены. Команда `\-`, указывающая перенос в словах, в этих скобках не нужна. Для расстановки акцентов над символами в скобках `\tabbing` используются команды `\a=`, `\a'`, `\a'`.

Командные скобки `tabbing` не могут быть вложенными одна в другую.

Некоторые единицы измерения в L^AT_EX'e.

mm	<i>миллиметр</i>	▬
pc	<i>пика</i>	┌┐
сс	<i>цицера</i>	┌┐
cm	<i>сантиметр</i>	┌┐┌┐
in	<i>дюйм</i>	┌┐┌┐┌┐

```

\newcommand{\rv}{\rule{.4pt}{4pt}}
\newcommand{\rvb}{\rv\hrulefill\rv}
Некоторые единицы измерения в \LaTeX'e.
\begin{tabbing}
mmmmmm\= mmmmmmmmmmmmm\=\hspace{3cm}
\= \kill
mm\>{\it миллиметр}\>\makebox[1mm]{\rvb}\\
pc\>{\it пика}\>\makebox[1pc]{\rvb}\\
cc\>{\it пиперо}\>\makebox[1cc]{\rvb}\\
cm\>{\it сантиметр}\>\makebox[1cm]{\rvb}\\
in\>{\it дюйм}\>\makebox[1in]{\rvb}\\
\end{tabbing}

```

Как меняется вид строки при применении контрольных символов, без ввода нового шаблона.

a	b	c	d	e
Строки изменены плюсами				
		aa	bb	cc
	dd	ee	ff	gg
		aa	bb	cc
	bb	cc	ee	
Шаблон восстановлен				
a	b	c	d	e
Элементы со значками				
á	ḃ	ċ	d	e

```
\begin{tabbing}
111111\=112222\=113333\=114444\=111555
\kill\\
a \> b \> c \> d \> e \+ \+ \\
\' Строки изменены плюсами \\
aa\> bb\> cc\\ \<dd \>ee \>ff \>gg \\
aa \> bb \> cc \- \\
\' bb \> cc \>ee \- \\
\' Шаблон восстановлен \\
a \> b \> c \> d \> e \\
\> \' Элементы со значками\\
\'a\'{a}\> \'a\'{b}\> \'a\'{c} \>d \>e\\
\end{tabbing}
```

16.2. Построение произвольных таблиц

Командные скобки `tabular` создают бокс из последовательности рядов, элементы которых выровнены вертикально в столбцы.

Символ `&` является разделителем элементов ряда.

Команда `\\` есть признак конца строки. Эта команда `\\` может иметь необязательный параметр `<величина>`, изменяющий расстояние между этой и следующей строкой на заданную величину.

```
▷ \begin{tabular}[<положение>]{<столб.1>...<столб.n>}
    <вход 1> & <вход 2> & ... \\    [<величина>]
    .....
\end{tabular}
```

Скобки имеют следующие параметры.

- **<положение>**

Необязательный параметр, указывающий положение таблицы по вертикали относительно текущей строки: в строке располагается верх таблицы (`t`), таблица центрирована относительно текущей строки (`c`), в строке располагается низ таблицы (`b`). По умолчанию таблица центрируется.

- **<столб.1>...<столб.n>**

Обязательный параметр, указывающий способ форматирования столбцов. Для каждого столбца должно быть задано одно из приведенных ниже значений:

`l` – выравнивание столбца по левому краю;

`r` – выравнивание столбца по правому краю;

`c` – выравнивание столбца относительно центра;

`|` – столбец состоит из вертикальной линии на полную высоту и глубину таблицы;

`@{text}` – вставка текста в каждый ряд таблицы. @-выражение подавляет обычный промежуток между столбцами, поэтому любой промежуток между вставленным текстом и смежными элементами должен быть включен в текст явно. В @-выражении может использоваться команда `\hspace*`. Только в @-выражении может использоваться команда `\extracolsep{<ширина>}`, вводящая заданное в ней дополнительное пространство слева во все последующие столбцы.

`p{wd}` – создает столбец, в котором каждый элемент печатается как абзац шириной `wd`. Команда `\\` не может появляться в подобном элементе за исключением следующих ситуаций: (i) внутри командных скобок, подобных `minipage`, `array` или `tabular`; (ii) в параметре `<текст>` команд `\parbox`, `\centering`, `\raggedright`, или `\raggedleft`.

Ниже приведены команды, используемые в пределах командных скобок `tabular`.

```
▷ \cline{i-j}
```

Рисует горизонтальные линии через столбцы, номера которых указаны в обязательном параметре.

▷ `\hline`

Рисует горизонтальную линию по ширине таблицы. Может быть выдана до, после или между строками массива.

▷ `\multicolumn{<столбцы>}{<положение>}{<текст>}`

Используется для объединения нескольких элементов строки в один. Все параметры обязательны. Первый указывает число столбцов для объединения. Второй указывает на наличие или отсутствие вертикальных линий и тип выравнивания объединенного столбца: относительно центра — (c), левого края — (l), правого края — (r).

Третий параметр задает текст.

Эта команда должна либо начинать строку, либо следовать сразу же за символом разделения элементов &.

▷ `\vline`

Рисует вертикальную линию на полную высоту и глубину строки. Для сдвига линии в пределах столбца можно использовать команду `\hfill`. Команда `\hfill` может также использоваться в @-выражении.

A B C
0 1 2
S T U

A	B	C
0	1	2
S	T	U

```
{\large\vspace*{1cm}\begin{tabular}{ccc}
A & B & C\\ 0&1&2\\ S & T & U\\
\end{tabular} \hspace*{1.5cm}
\begin{tabular}{|ccccc|}
\hline A & & B & & C \\ \hline
0 & \vline & 1 & \vline & 2\\
\hline S & & T & & U \\ \hline
\end{tabular}}
```

Некоторые стилевые файлы построения библиографии	
agsm.bst	Australian Government publications
astron.bst	Astronomy Bib _{TeX} style
cbe.bst	Council of Biology Editors style (включает такие журналы как American Naturalist, Evolution).
humannat.bst	Стиль для журналов Human Nature, American Antropologist
phapalik.bst	American psychological Association Style
phrmp.bst	Review of Modern Physics

```
\begin{tabular}{|l|p{8cm}|}
```

```
\hline
```

```
agsm.bst      & Australian Government publications \\
astron.bst    & Astronomy Bib\TeX\ style \\
cbe.bst       & Council of Biology Editors style (включает
```

```

        такие журналы как    American Naturalist,
        Evolution).\
humannat.bst& Стиль для журналов Human Nature,
        American Antropologist\
phapalik.bst& American psychological Association Style \
phrmp.bst   & Review of Modern Physics \
\hline
\end{tabular}

```

16.3. Как построить таблицу, длина которой превышает размер страницы

Для печати таблиц, длина которых превышает размер страницы, существует несколько корректирующих стилей, полная информация о которых имеется в описании библиотеки корректирующих стилей.

Например, в списке корректирующих стилей команды `\documentstyle` можно указать стиль `supertab`.

При работе с этим стилем прежде всего специальными командами задается информация для оформления заголовка и концовки таблицы на каждой странице.

- ▷ `\tablefirsthead{<текст>}`
Определяет первый заголовок таблицы. Команда не является обязательной. Параметр должен оканчиваться символами `\` и может быть задан в виде строки таблицы.
- ▷ `\tablehead{<текст>}`
Определяет заголовок таблицы на всех страницах, т.е. и на первой. Заголовок должен оканчиваться символами `\`.
- ▷ `\tabletail{<текст>}`
Определяет концовку таблицы на всех страницах.
- ▷ `\tablelasttail{<текст>}`
Определяет дополнительную концовку таблицы на последней странице. Команда не является обязательной. Может быть задана в виде строки таблицы.
- ▷ `\topcaption{<текст>}`
Определяет общий заголовок супертаблицы, который будет помещен перед ней.
- ▷ `\bottomcaption{<текст>}`
Определяет общий заголовок супертаблицы, который будет помещен в конце ее.
- ▷ `\tablecaption{<текст>}`
Определяет общий заголовок супертаблицы, который будет помещен в соответствии с принятым по умолчанию стандартом — над таблицей.

Командные скобки построения длинной таблицы имеют формат, совпадающий с форматом командных скобок для обычных таблиц, и используют те же параметры.

```

▷ \begin{supertabular}{<столб.1>...<столб.n>}
    <вход 1> & <вход 2> & ... \\    [<величина>]
    .....
\end{supertabular}

```

Существуют и другие стили, позволяющие вводить длинные таблицы и массивы. Эти стили не всегда совместимы между собой, поэтому при работе с ними следует быть осторожным и умеренным оптимистом.

Таблица 3. Очень большая таблица

Номер	Пункт1	Пункт2	Пункт3	Пункт4
1	aaa	bbbb	cccc	dddd
2	aaa	bbbb	cccc	dddd
3	aaa	bbbb	cccc	dddd
4	aaa	bbbb	cccc	dddd
5	aaa	bbbb	cccc	dddd
6	aaa	bbbb	cccc	dddd
7	aaa	bbbb	cccc	dddd
8	aaa	bbbb	cccc	dddd
9	aaa	bbbb	cccc	dddd
10	aaa	bbbb	cccc	dddd
11	aaa	bbbb	cccc	dddd
12	aaa	bbbb	cccc	dddd
13	aaa	bbbb	cccc	dddd
14	aaa	bbbb	cccc	dddd
15	aaa	bbbb	cccc	dddd
16	aaa	bbbb	cccc	dddd
17	aaa	bbbb	cccc	dddd
18	aaa	bbbb	cccc	dddd
19	aaa	bbbb	cccc	dddd
20	aaa	bbbb	cccc	dddd
21	aaa	bbbb	cccc	dddd
22	aaa	bbbb	cccc	dddd
23	aaa	bbbb	cccc	dddd
24	aaa	bbbb	cccc	dddd
25	aaa	bbbb	cccc	dddd
26	aaa	bbbb	cccc	dddd
27	aaa	bbbb	cccc	dddd
28	aaa	bbbb	cccc	dddd
29	aaa	bbbb	cccc	dddd

Номер	Пункт1	Пункт2	Пункт3	Пункт4
30	aaa	bbbb	cccc	dddd
31	aaa	bbbb	cccc	dddd
32	aaa	bbbb	cccc	dddd
33	aaa	bbbb	cccc	dddd
34	aaa	bbbb	cccc	dddd
35	aaa	bbbb	cccc	dddd
36	aaa	bbbb	cccc	dddd
37	aaa	bbbb	cccc	dddd
38	aaa	bbbb	cccc	dddd
39	aaa	bbbb	cccc	dddd
41	aaa	bbbb	cccc	dddd
42	aaa	bbbb	cccc	dddd
43	aaa	bbbb	cccc	dddd
44	aaa	bbbb	cccc	dddd
45	aaa	bbbb	cccc	dddd
46	aaa	bbbb	cccc	dddd
47	aaa	bbbb	cccc	dddd
48	aaa	bbbb	cccc	dddd
49	aaa	bbbb	cccc	dddd
50	aaa	bbbb	cccc	dddd

END

```

\tablecaption{Очень большая таблица}
\tablefirsthead{\hline Номер & Пункт1 & Пункт2 & Пункт3 & Пункт4 \\ \hline }
\tablehead{\hline Номер & Пункт1 & Пункт2 & Пункт3 & Пункт4 \\ \hline }
\tabletail{\hline}
\tablelasttail{\hline \multicolumn{5}{c}{END}\\ }
\begin{supertabular}{|l|c|c|c|c|}
1 & aaa & bbbb & cccc & dddd \\
.....
50 & aaa & bbbb & cccc & dddd \\
\end{supertabular}

```

17. Построение диаграмм

В ЛАТ_ΕX'е имеются командные скобки для построения рисунка-диаграммы, который может содержать текст, линии, стрелки, окружности и овалы:

▷ `\begin{picture}(<ширина>,<высота>)(<х смещение>,<у смещение>)`
 <тело рисунка>
`\end{picture}`

Эти командные скобки имеют обязательный параметр (`<ширина>`, `<высота>`), задаваемый в *круглых* скобках. Он определяет ширину и высоту отводимого для рисунка бокса в единицах `unitlength`, по умолчанию равных `1pt`.

Необязательный параметр, который меняет положение нижнего левого угла этого бокса, задается также в круглых скобках.

Значение единицы длины `unitlength` можно изменить командой `\setlength`, но если попытаться менять их внутри командных скобок `picture`, результаты могут быть чрезвычайно странными.

<Тело рисунка> состоит из набора команд `\put`, размещающих отдельные элементы рисунка:

▷ `\put(x,y){<объект>}`

Команда помещает объект, определенный во втором параметре, в точку (x,y) . Объектом является текст, могущий содержать команды, например, команды описания графических объектов или другие командные скобки `picture`. Координата указывает длину в величинах, кратных единице длины `unitlength`. Координаты задаются по отношению к начальной точке, которая находится в нижнем левом углу картинки, и должны быть заключены в круглые скобки. Координаты указывают длину в величинах, кратных единице длины `unitlength`.

Ниже описаны команды различных графических элементов, допустимых внутри командных скобок `picture`.

Графические объекты

▷ `\circle{<диаметр>}`

Создает окружность, диаметр которой задан в единицах `unitlength`. ЛАТ_ΕX имеет фиксированный набор окружностей и выберет ту, диаметр которой наиболее близок к заданному.

▷ `\circle*{<диаметр>}`

Создает заштрихованную окружность, диаметр в единицах `unitlength`. ЛАТ_ΕX имеет фиксированный набор окружностей и выберет ту, диаметр которой наиболее близок к заданному.

▷ `\dashbox{<размер пунктира>}<ширина>,<высота>[<положение>]{<текст>}`

Создает пунктирную рамку вокруг параметра `<текст>` заданной ширины и высоты (в принятых для рисунка единицах длины). Длина пунктирной черточки выбирается по

умолчанию, однако можно явно указать ее в первом параметре в виде целого положительного числа. <Текст> может включать в себя команды. Если требуется пустая рамка, этот параметр можно опустить. Необязательный параметр <положение> указывает на квадрант, в котором появляется текст. Можно выбрать один или два совместимых параметра из следующих: верх бокса — (t), — низ бокса (b), левый край — (l), правый край — (r).

По умолчанию текст останется в центре бокса.

▷ `\frame{<текст>}`

Вставляет в прямоугольную рамку указанный в параметре текст. Точка привязки находится в левом нижнем углу рамки. Никакого дополнительного промежутка между рамкой и объектом не возникает.

▷ `\framebox(<ширина>,<высота>)[<положение>]{<текст>}`

Строит и обводит сплошной рамкой бокс указанной ширины и высоты. <Текст> может включать в себя команды. Если требуется пустая рамка, этот параметр можно опустить. Необязательный параметр <положение> указывает на квадрант, в котором появляется текст. Можно выбрать один или два совместимых параметра из следующих: верх бокса — (t), низ бокса — (b), левый край — (l), правый край — (r).

По умолчанию текст останется в центре бокса.

▷ `\line(<х-наклон>,<у-наклон>){<длина>}`

Рисует линию указанной длины и наклона. Определяющие наклон значения <х-наклон>, <у-наклон> должны лежать между -4 и +4, длина задается в единицах `unitlength`.

▷ `\linethickness{<размерная величина>}`

Устанавливает толщину горизонтальных и вертикальных линий внутри командных скобок `picture`. Не влияет на толщину наклонных линий и окружностей или четвертей окружностей, нарисованных с помощью команды `\oval`.

▷ `\makebox(<ширина>,<высота>)[<положение>]{<текст>}`

Строит бокс указанной ширины и высоты. Необязательный параметр <положение> указывает на квадрант, в котором появляется текст. Можно выбрать один или два совместимых параметра из следующих: верх бокса — (t), низ бокса — (b), левый край — (l), правый край — (r).

По умолчанию текст останется в центре бокса.

Параметры, определяющие размер бокса, не являются обязательными. Если они не указаны, размер бокса определяется помещаемым в него текстом, но при этом точка, заданная в команде `\put`, будет находиться в центре бокса. Для помещения ее в нижний левый угол бокса, размер которого должен определяться текстом, первый параметр должен быть задан как (0,0).

▷ `\oval(<ширина>,<высота>)[<часть>]`

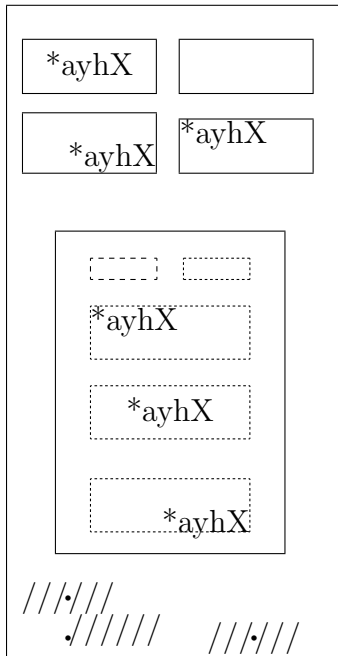
Создает прямоугольник с закругленными углами. Необязательный параметр <часть> разрешает выбрать часть овала: верхняя половина — (t), нижняя половина — (b), правая половина — (r), левая половина — (l).

Комбинации `tr`, `tl`, `br` и `bl` выбирают соответствующие четверти овала.

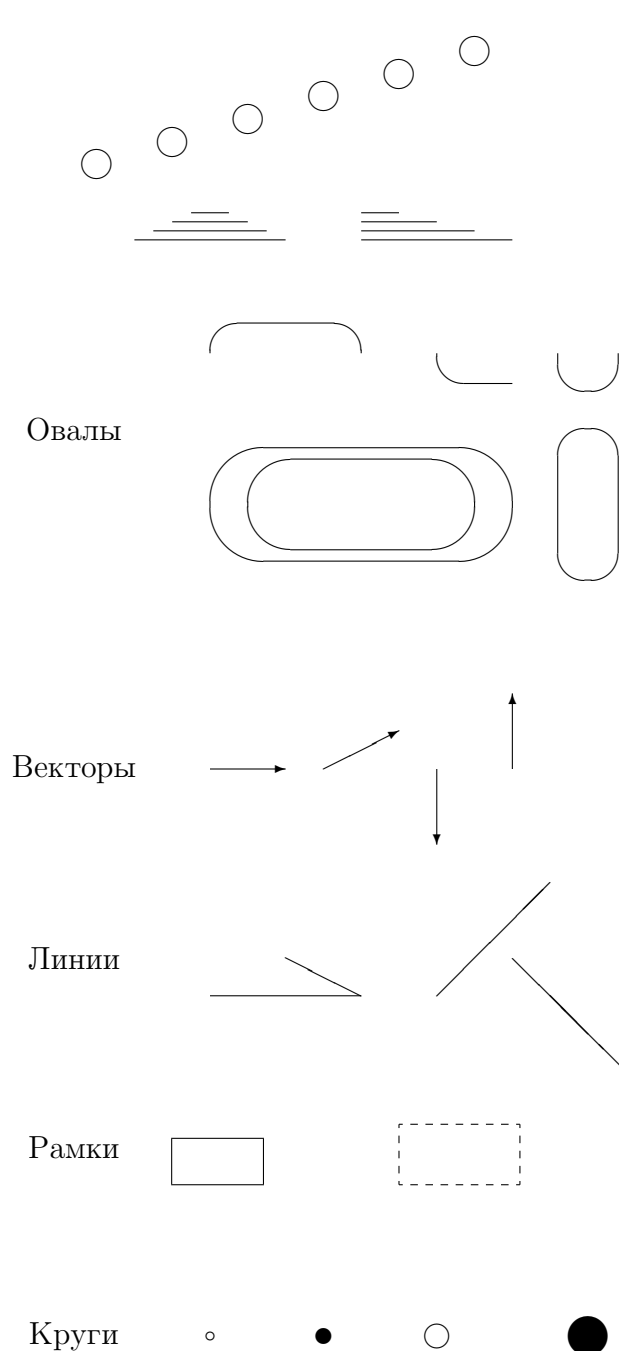
Координаты в команде `\put` определяют центр овала.

- ▷ `\shortstack[<положение>]{<вход 1> \\ <вход 2> \\ <вход n>}`
Создает бокс с одним столбцом объектов с опорной точкой в нижнем левом углу. Элементы столбца разделяются командой `'\\'`. Тип выравнивания элементов задается в необязательном параметре: по правому краю (`r`), по левому краю (`l`), центрированно (`c`). По умолчанию элементы в столбце центрируются.
- ▷ `\vector(<x-наклон>,<y-наклон>){<длина>}`
Рисует линию со стрелкой указанной длины и наклона. Определяющие наклон значения координат `<x-наклон>`, `<y-наклон>` должны лежать между -4 и +4.
- ▷ `\multiput(x,y)(delta-x,delta-y){<кол-во>}{<объект>}`
Располагает регулярным способом заданное количество копий объекта. Первый параметр задает положение первого объекта, второй — смещение каждой последующей копии по отношению к предыдущей.

Мы приводим здесь два примера. Первый пример показывает, где помещается точка привязки при разных форматах кодады `\put`, как можно размещать текст в боксах, как работают в пределах скобок `picture` другие кодадные скобки `picture`. Второй показывает все виды элементарных графических объектов, которые может строить $\text{\LaTeX}$.



```
\fbox{\begin{picture}(120,240)
  \put(20,5){\circle*{2}}
  \put(20,20){\circle*{2}}
  \put(90,5){\circle*{2}}
  \put(20,5){\makebox{//////}}
  \put(20,20){\makebox(0,0){//////}}
  \put(90,5){\makebox(0,0){//////}}
  %
  \put(15,40){\fbox{\begin{picture}(80,115)
    \put(10,5){\dashbox{1}(60,20)[br]{*ayhX}}
    \put(10,40){\dashbox{1}(60,20)[c]{*ayhX}}
    \put(10,70){\dashbox{1}(60,20)[t1]
                                     {*ayhX}}
    \put(10,100){\dashbox{2}(25,8)}
    \put(45,100){\dashbox(25,8)}
  \end{picture}}}
  %
  \put(3,180){\framebox(50,20)[br]{*ayhX}}
  \put(62,180){\framebox(50,20)[t1]
                                     {*ayhX}}
  \put(3,210){\framebox(50,20)[c]{*ayhX}}
  \put(62,210){\framebox(50,20)}
\end{picture}}
```



Объекты диаграмм.

```

\unitlength=1mm
\begin{picture}(110,200)
\put(15,2){\makebox
{\Large \sl Объекты диаграмм.}}
%
\put(2,25){\makebox(0,0){Круги}}
\put(20,25){\circle{1}}
\put(35,25){\circle*{2}}
\put(50,25){\circle{3}}
\put(70,25){\circle*{6}}
%
\put(2,50){\makebox(0,0){Рамки} }
\put(15,45){\framebox(12,6)[c]{\ }}
\put(45,45){\dashbox(16,8)[c]{\ }}
%
\put(2,75){\makebox(0,0){Линии}}
\put(20,70)
{\line(1,0){20}\line(-2,1){10}}
\put(50,70)
{\line(1,1){15}\line(1,-1){10}}
\put(60,75){\line(1,-1){10}}
%
\put(2,100){\makebox(0,0){Векторы}}
\put(20,100){\vector(1,0){10}}
\put(35,100){\vector(2,1){10}}
\put(50,100){\vector(0,-1){10}}
\put(60,100){\vector(0,1){10}}
%
\put(2,145){\makebox(0,0){Овалы}}
\put(40,135)
{\oval(30,12)\oval(40,15)}
\put(70,135){\oval(8,20)}
\put(30,155){\oval(20,8)[t]}
\put(60,155){\oval(20,8)[b]}
\put(70,155){\oval(8,10)[b]}
%
\put(10,170){\shortstack{
\line(1,0){5}\ \ \line(1,0){10}\ \
\line(1,0){15}\ \ \line(1,0){20}}}
\put(40,170){\shortstack[l]{
\line(1,0){5}\ \ \line(1,0){10}\ \
\line(1,0){15}\ \ \line(1,0){20}}}
%
\multiput(5,180)(10,3){6}{\circle{4}}
\end{picture}

```

18. Ввод текста из нескольких файлов

Документ большого объема часто бывает удобнее держать не в одном исходном файле, а в нескольких. Однако когда каждая из частей отлажена, разумно, чтобы весь документ обрабатывался при запуске одного корневого файла, содержащего команды использования внешних файлов.

▷ `\input {<имя файла>}`

Вызывает указанный файл, чтобы прочитать и обработать, точно как, если его содержимое было бы помещено в текущий файл в данный момент. Имя файла может быть полное имя файла с расширением или простое, в случае, когда файл имеет расширение `.tex`.

▷ `\include {<имя файла>}`

Указывает файл, который следует включить в обрабатываемый текст. Может быть использована в сочетании с командой `\includeonly` для выборочного включения файлов. Если файл, определенный командой `\include`, задан в списке параметров команды `\includeonly`, или если последняя не используется вовсе, то команда `\include` эквивалентна командам

```
\newpage \input{file} \newpage
```

Если файл не значится в списке файлов команды `\includeonly`, то `\include` эквивалентна команде `\newpage`.

Команда `\include` не может появиться в преамбуле или в файле, читаемом другой командой `\include`. Если файл, указанный в `\include` и `\includeonly`, не существует, будет выдано предупреждение.

▷ `\includeonly {<список файлов>}`

Содержит список файлов, которые будут прочитаны командой `\include`. Команда может появляться только в преамбуле.

Названия файлов в списке разделяются запятыми.

19. Разбиение на строки и страницы

Обработку текста L^AT_EX начинает с преобразования исходного файла в ряд глифов (специфического представления символов) и промежутков между ними. Далее этот ряд должен быть разбит на строки, а строки объединены в отдельные страницы. В основном L^AT_EX это делает автоматически, используя параметры файлов *описания стиля*, но существуют команды, с помощью которых пользователь может формировать строки и страницы сам.

▷ `\[<дополнит. промежуток>]`

▷ `\*[<дополнит. промежуток>]`

Команда перехода на новую строку. Имеет необязательный параметр, определяющий дополнительный вертикальный промежуток, который должен быть вставлен перед новой строкой. Параметр может быть отрицательным.

Команда `\*` подобна команде `\[`, но запрещает L^AT_EX'у начинать после этой строки новую страницу.

▷ `\-`

Разрешает переносить слово в данном месте. Полезна для указания возможности переноса в словах, содержащих дефис.

▷ `\cleardoublepage`

В стиле двусторонней печати заканчивает страницу и выводит плавающие элементы: рисунки и таблицы — для которых не нашлось подходящего места в тексте. Последующую страницу эта команда делает правосторонней нечетной, создавая пустую страницу, если необходимо.

▷ `\clearpage`

Заканчивает страницу и выводит плавающие элементы: рисунки и таблицы — для которых не нашлось подходящего места в тексте.

▷ `\hyphenation{<слова>}`

Указывает точки переноса в списке слов. Слова отделяются друг от друга пробелами, точки переноса указываются с помощью дефиса.

▷ `\linebreak[<число>]`

Указывает место разбиения строки. Необязательный параметр `<число>`, определяющий степень необходимости такого разбиения, может быть числом от 0 до 4. Большее число есть более настоятельная просьба. Команда `\linebreak` заставляет L^AT_EX растянуть оставшуюся часть строки так, чтобы она простиралась до правого поля.

▷ `\newline`

Выполняет переход на новую строку.

▷ `\newpage`

Выполняет переход на новую страницу.

▷ `\nolinebreak[<число>]`

Запрещает разрывать текущую строку в данном месте. Необязательный параметр `<число>`, определяющий степень категоричности этого требования, может быть числом от 0 до 4. Большее число есть более настоятельная просьба.

▷ `\nopagebreak[<число>]`

Запрещает разрывать текущую страницу в месте выдачи команды. Необязательный параметр `<число>`, определяющий степень категоричности этого требования, может быть числом от 0 до 4. Большее число есть более настоятельная просьба.

▷ `\pagebreak[<число>]`

Указывает место разрыва текущей страницы. Необязательный параметр `<число>`, определяющий степень категоричности этого требования, может быть числом от 0 до 4. Большее число есть более настоятельная просьба.

Во многих случаях полезен ввод мини-страницы.

Правила разбиения строк на части и перехода на новую страницу даны на странице 61, в разделе 19.

Во многих случаях полезен ввод мини-страницы.

Правила разбиения строк на части и перехода на новую страницу даны на странице 61, в разделе 19.

Во многих случаях полезен ввод мини-страницы.

`\par` Правила разбиения строк на части и перехода на новую страницу даны на странице `\pageref{break}`, в разделе `\ref{break}`.

`\par\bigskip`

Во многих случаях полезен ввод мини-страницы.

`\par` Правила разбиения строк `\linebreak` на части и перехода на новую страницу даны на странице `\pageref{break}`, в разделе `\ref{break}`.

20. Стили страниц, титульная информация

Стиль страницы определяет все атрибуты оформления страницы. Важнейшие из них:

- максимальные высота и ширина текста;
- расположение текста — одноколонный, в две колонки и прочее;
- тип нумерации — арабскими или римскими цифрами, буквами;
- расположение номеров страниц — по центру или иначе, вверху или внизу;
- наличие и формат колонтитулов — «бегущих» заголовков, воспроизводящих на каждой странице заголовок текущей рубрики.

По умолчанию во всех базовых стилях страница оформляется по формату Plain T_EX: текст печатается в одну колонку, номера страниц заданы арабскими цифрами и располагаются внизу по центру и всегда на одном уровне (если текст, например, в конце главы, не занимает страницу целиком, добавляется надлежащий вертикальный промежуток), заголовок текущей рубрики над телом страницы не выдается.

Пользователь, однако, имеет в своем распоряжении средства для изменения стиля отдельной страницы или группы страниц.

▷ `\pagestyle{<стиль страницы>}`

Меняет стиль страницы, начиная с текущей и до конца документа или новой команды `\pagestyle`.

Допустимые стили страниц:

- | | |
|-------------------------|--|
| <code>plain</code> | страница оформляется по формату Plain T _E X (принято по умолчанию); |
| <code>empty</code> | номера страниц не проставляются; |
| <code>headings</code> | над телом каждой страницы помещается колонтитул, воспроизводящий заголовок текущей рубрики; |
| <code>myheadings</code> | помещается колонтитул, текст которого пользователь определил с помощью команд <code>\markboth</code> или <code>\markright</code> . |

▷ `\thispagestyle{<стиль страницы>}`

Команда работает с теми же параметрами выбора, что и команда `\pagestyle`, но меняет стиль только для текущей страницы.

▷ `\onecolumn`

Начинает новую страницу и приводит к печати в одну колонку дальнейших страниц текста (принято по умолчанию).

▷ `\twocolumn`

Начинает новую страницу и приводит к печати в две колонки дальнейших страниц текста.

▷ `\flushbottom`

Делает все страницы текста одинаковой высоты, добавляя вертикальное пространство, если текст не занимает страницу целиком (принято по умолчанию).

▷ `\raggedbottom`

Делает высоту каждой страницы по высоте текста на этой странице, не выравнивая, таким образом, вертикальный размер страниц, не заполненных текстом до конца.

▷ `\pagenumbering{<стиль>}`

Указывает стиль нумерации страниц. Возможные значения параметра:

`arabic` нумерация арабскими цифрами;
`roman` нумерация малыми римскими цифрами;
`Roman` нумерация большими римскими цифрами;
`alph` нумерация строчными буквами;
`Alph` нумерация прописными буквами.

Если в команде `\documentstyle` указан корректирующий стиль `russian`, при нумерации страниц буквами будут использоваться буквы русского алфавита.

▷ `\markboth{<текст слева>}{<текст справа>}`

▷ `\markright{<текст справа>}`

Используются в сочетании со стилем страницы `myheadings` для установки двустороннего или правостороннего колонтитула.

▷ `\begin{multicols}{<кол-во колонок>}`
`<текст>`
`\end{multicols}`

Командные скобки построения фрагмента текста в несколько колонок. Обязательный параметр указывает число колонок. Для того, чтобы $\text{\LaTeX}$ распознавал эти командные скобки, в первой команде документа следует указать корректирующий стиль `multicol` (обратите внимание на то, что в названии стиля отсутствует последняя буква названия командных скобок, а в остальном эти названия совпадают).

Стиль страницы определяет расположение текста на странице, формат колонтитулов («бегущих» заголовков), тип нумерации и прочее.

По умолчанию во всех базовых стилях страница оформляется по формату Plain $\text{\TeX}$: текст печатается в одну колонку, номера страниц печатаются араб-

скими цифрами располагаются внизу по центру и всегда на одном уровне, колонтитул — бегущий заголовок текущей рубрики над телом страницы — не выдается.

Однако пользователь имеет в своем распоряжении средства, позволяющие изменять в определенных пределах стиль одной страницы или группы страниц.

```
\documentstyle[12pt,russian,multicol]{article}
```

```
.....
```

```
\begin{multicols}{3}
```

Стиль страницы определяет расположение текста на странице, формат колонтитулов (<бегущих> заголовков), тип нумерации и прочее. \par По умолчанию во всех базовых стилях страница оформляется по формату \PT: текст печатается в одну колонку, номера страниц печатаются арабскими цифрами располагаются внизу по центру и всегда на одном уровне, колонтитул --- бегущий заголовок текущей рубрики над телом страницы --- не выдается.

```
\end{multicols}
```

Однако пользователь имеет в своем распоряжении средства, позволяющие изменять в определенных пределах стиль одной страницы или группы страниц.

Титульная страница

В базовых стилях L^AT_EX'a титульная информация формируется автоматически. В стиле **article** эта информация помещается над началом текста, в стилях **report** и **book** — на отдельной титульной странице. Для задания титульной информации для всех трех стилей имеются четыре команды, как правило, задаваемые в преамбуле документа.

▷ **\author{<имена>}**

Указывает автора или список авторов, имена которых должны разделяться командами **\and**. Используйте команду **** для разделения строк внутри каждого входа в этом списке, например, чтобы указать место работы автора или адрес.

▷ **\title{<текст>}**

Определяет заголовок. Используйте команду **** для разделения строк в длинном заголовке.

▷ **\date{<дата>}**

Определяет дату выдачи документа. Если команда опущена, будет выдана текущая дата. При пустом параметре выдача даты не производится.

▷ **\thanks{<текст>}**

Создает сноску к заголовку.

Ввод титульных данных осуществляется командой, которую надо задать сразу после **\begin{document}**.

▷ **\maketitle**

Начинает новую страницу и вводит титульную информацию в соответствии со стандартами заданного базового стиля.

Если титульные данные — заголовок и авторы — ранее не определены будет выдано сообщение об ошибке.

Пользователь может построить собственную титульную страницу. Командные скобки построения титульной страницы, т.е. страницы без номера, формирование которой предоставляется пользователю, имеют формат:

—

▷ `\begin{titlepage}`
 `<текст>`
`\end{titlepage}`

Страница, следующая за титульной, всегда имеет номер 1.

Если на титульной странице должна стоять дата выдачи документа , ее следует ввести командой

▷ `\today`

Дата последней обработки L ^A T _E X'ом текста нашей книги: 16 января 1999.	Дата последней обработки \L'ом текста нашей книги: <code>\today</code> .
---	--

21. Вывод сообщений на терминал

Для вывода сообщений на терминал имеются две команды:

▷ `\typeout{<сообщение>}`

Печатает сообщение на терминале и в протольном файле. Правило ЛАТ_ЕХ’а относительно ввода пробелов соблюдается и для этих сообщений: множество пробелов рассматриваются как один, а пробелы после команды не печатаются. Для вывода символа пробела можно использовать команду `\space`.

▷ `\typein[<имя>]{<сообщение>}`

Печатает сообщение на терминале и останавливается в ожидании строки ввода с терминала.

Если необязательный параметр опущен, то ввод строки вставляется в документ в месте появления команды `\typein` как обычный фрагмент текста.

Если же этот параметр задан, то он считается названием команды, определение которой задается введенной строкой. Определенная таким образом команда ничем не отличается от команд, определенных посредством `\newcommand`.

22. Математические выражения

Математические выражения обрабатываются L^AT_EX'ом в особом режиме работы, называемом математической модой. Общий для T_EX'a и L^AT_EX'a контрольный символ перехода в эту моду есть знак доллара \$.

В научных статьях и книгах короткие простые (с полиграфической точки зрения) математические выражения часто печатаются непосредственно в строке текста.

Математические выражения в строке текста должны заключаться в одинарные знаки доллара: \$.\$.

Математические выражения, печатаемые отдельной выделенной строкой, должны заключаться в двойные знаки доллара: \$\$. \$\$.

В L^AT_EX'е имеются дополнительные средства перехода в математическую моду. Это командных скобки:

`math` `displaymath`

Командные скобки `math` (эквивалент \$.\$.) существуют для формул, которые являются частью текстовой строки. Командные скобки `displaymath` (эквивалент \$\$. \$\$.) — для формул, которые печатаются отдельной строкой. Для скобок `math` и `displaymath` имеется еще одна сокращенная форма записи, отличная от ограничителей в виде одинарных или двойных знаков доллара:

`\(...\)` для `\begin{math}...\end{math}`
`\[...\]` для `\begin{displaymath}...\end{displaymath}`

Вводимые в строке текста математические выражения могут быть разорваны между строками при компоновке текста L^AT_EX'ом. Выражения, вводимые отдельной строкой, не разрываются, даже если они превышают ширину страницы.

Команда математической моды, выданная при обработке простого текста (L^AT_EX находится в моде абзаца) вызовет остановку в работе с выдачей сообщения об ошибке.

22.1. О размерах и стиле символов в формулах

При построении математических выражений используются символы трех размеров: основного, для индексов первого уровня и для индексов остальных уровней.

Если в выражении имеются верхние и(или) нижние индексы, сами являющиеся индексированными величинами, то при основном размере шрифта документа 10 pt размер индексов первого уровня будет 7 pt, а индексов второго и высших уровней — 5 pt.

Пользователь может, однако, строить нестандартные (для L^AT_EX'a) формулы, указывая в них явно размеры символов следующими командами T_EX'a.

▷ `\textstyle <символ>`

Команда выбирает основной размер символов для формул в текстовой строке.

▷ `\displaystyle <символ>`

Команда выбирает основной размер символов для формул, выводимых отдельной строкой.

▷ `\scriptstyle <символ>`

Команда выбирает размер, определенный для индексов первого уровня.

▷ `\scriptscriptstyle <символ>`

Команда выбирает размер, определенный для индексов высших уровней.

$n + n + n$	<code>\$\$n+\scriptstyle n+</code>
	<code>\scriptscriptstyle{n}\$\$</code>
$n + n + n$	<code>{\LARGE \$\$n+ \scriptstyle n+</code>
	<code>\scriptscriptstyle{n}\$\$}</code>

По умолчанию буквы печатаются в математических формулах курсивом. Если желательно выдать их другим шрифтом, следует явно указать тип шрифта.

Помещение символов в бокс, созданный одной из команд построения боксов приведет к печати их прямым шрифтом.

Для заглавных букв в математической моде можно использовать каллиграфический шрифт. Этот шрифт выбирается командой `\cal`.

$N + N + N + N$	<code>\$\$N+{\rm N}+{\sf N}+{\sl N}\$\$</code>
$N + \boxed{N}$	<code>\$\$ \mbox{N}+\fbox{N}\$\$</code>
$ABCDIFGHIJKLMNOPQR$	<code>\$\${\cal ABCDIFGHIJKLMNOPQR}\$\$</code>
$STUVWXYZ$	<code>\$\${\cal STUVWXYZ}\$\$</code>

22.2. Пробелы в математической моде

Пробелы, вводимые простым нажатием клавиши, в математической моде игнорируются.

Л^AT_EX устанавливает пробелы по правилам, принятым в полиграфии для математических выражений. Кроме того, в математической моде воспринимаются команды образования пробелов `\quad` и `\qquad`. Для введения в формулы специальных пробелов существуют следующие команды:

`\;` — большой пробел; `\,` — малый пробел;
`\:` — средний пробел; `\!` — отрицательный малый пробел.

1	1[1[1[1[1	$\$1\qquad 1[1[1[1[1\$ \backslash \text{par}$
1	1[1[1[1[1	$\$1\qquad 1 \quad [1 \quad [1 \quad [1 \quad [1\$ \backslash \text{par}$
1	1 [1 [1 [1 [1	$\$1\qquad 1\backslash; [1\backslash; [1\backslash; [1\backslash; [1\$ \backslash \text{par}$
1	1 [1 [1 [1 [1	$\$1\qquad 1\backslash: [1\backslash: [1\backslash: [1\backslash: [1\$ \backslash \text{par}$
1	1 [1 [1 [1 [1	$\$1\qquad 1\backslash, [1\backslash, [1\backslash, [1\backslash, [1\$ \backslash \text{par}$
1	1[1[1[1[1	$\$1\qquad 1\backslash! [1\backslash! [1\backslash! [1\backslash! [1\$$

22.3. Построение многоточий различного типа

- ▷ `\cdots`
Создает горизонтальные многоточия, точки расположены по центру строки.
- ▷ `\ddots`
Создает многоточия по диагонали.
- ▷ `\ldots`
Создает многоточия вдоль базовой линии строки. Команда работает в любой моде, не только математической.
- ▷ `\vdots`
Создает многоточия по вертикали.

$$a + \cdots + b + \ddots + c + \ldots + d + \vdots \qquad \text{\$}a+\backslash\text{cdots}+b+\backslash\text{ddots}+c+\backslash\text{ldots}+d+\backslash\text{vdots}\text{\$}$$

22.4. Верхние и нижние индексы

Чтобы получить выражение в виде нижнего индекса, перед ним следует набрать символ `_`. Чтобы получить выражение в виде верхнего индекса, перед ним следует набрать символ `^`. $\text{\LaTeX}$ допускает двойное и многократное индексирование. Неоднозначные комбинации индексов, например `A_2_x`, считаются ошибочными.

Для помещения индекса над символом имеется следующая команда:

- ▷ `\stackrel{<индекс>}{<символ>}`

Помещает индекс, указанный в первом параметре, над символом, указанным во втором параметре.

В формулах, вынесенных в отдельную строку, индекс над или под выражением можно поместить командой `\mathop` (команда $\text{\TeX}$ 'а):

- ▷ `\mathop {<символ>}^<индекс>`
- ▷ `\mathop {<символ>}_<индекс>`

X^n	X_n	X^{YZ}	$\$X^n \quad X_n \quad X^{Y^Z} \quad \$$
X_6^8	${}_4X^{YZ}$	${}_3X_5^1$	$\$X_{\{6\}}^{\{8\}} \quad {}_{\{4\}}X^{Y^Z} \quad {}^{\{2\}}_{\{3\}}X_{\{5\}}^{\{1\}} \quad X_{b^a} \quad X^{\{\displaystyle Y\}^{\textstyle Z}} \quad \$$
V^2	$\stackrel{a=const}{=}$	$2aS$	$\$V^2\backslash:\stackrel{a=const}{=}\backslash:2aS\quad \$$
X_*	X^*	ABC_{10}	$\{\mathop X_* \quad \mathop X^*\}\quad \{\mathop {ABC}_{10} \quad \mathop {abs}^*\}$
${}_1F^1$	${}_1F^1$		$\backslashmathop {abs}^*\}\quad \backslashhspace*{3cm} \quad \$_1F^1\quad \$_1F^1$

22.5. Символы со значками

x'	<code>x'</code>	x''	<code>x''</code>	$\bar{x}$	<code>\bar x</code>
$\hat{x}$	<code>\hat x</code>	$\vec{x}$	<code>\vec x</code>	$\dot{x}$	<code>\dot x</code>
$\ddot{x}$	<code>\ddot x</code>	$\check{x}$	<code>\check x</code>	$\grave{x}$	<code>\grave x</code>
$\breve{x}$	<code>\breve x</code>	$\acute{x}$	<code>\acute x</code>	$\tilde{x}$	<code>\tilde x</code>
$\imath$	<code>\imath</code>	j	<code>\jmath</code>		

Команды `\imath`, `\jmath` необходимы для построения выражений с различными значками над символами `i`, `j`.

$\vec{y} = \ddot{x} + \dot{x}$	$\bar{N} = (\hat{m}_i + \breve{m}_{i+1})/2$	$\$ \vec{y} = \ddot{x} \vec{\imath} + \dot{x} \vec{\jmath} \quad \$$
$\tilde{X}$	$\tilde{Y}$	$\$ \tilde{X} \quad \tilde{Y} \quad \$$
$\hat{Z}$		$\$ \hat{Z} \quad \$$

22.6. Надстрочные и подстрочные знаки для группы символов

- ▷ `\overbrace{<текст>}`
Помещает фигурную скобку над текстом.
- ▷ `\overline{<текст>}`
Проводит черту над текстом.
- ▷ `\underbrace{<текст>}`
Помещает фигурную скобку под текстом.

- ▷ `\underline{<текст>}`
Подчеркивает текст. Эта команда может использоваться в любой моде.
- ▷ `\widehat{<не более трех символов>}`
Помещает знак $\widehat{}$ над последовательностью символов.
- ▷ `\widetilde{<не более трех символов>}`
Помещает знак $\widetilde{}$ над последовательностью символов.

$$\overbrace{x+y+z} \quad \underbrace{xyz} \quad \widehat{xyz} \quad \widetilde{xyz} \quad \overline{x^{m+n+k}} \quad x^{\underline{n}}$$

$$\begin{aligned} & \text{\texttt{\$}\texttt{\$}\texttt{\backslash overbrace\{x+y+z\}\texttt{\backslash quad}} \\ & \text{\texttt{\backslash underbrace\{xyz\}\texttt{\backslash quad}}} \\ & \text{\texttt{\backslash widehat\{xyz\}\texttt{\backslash quad}\texttt{\backslash widetilde\{xyz\}\texttt{\$}\texttt{\$}}} \\ & \text{\texttt{\$}\texttt{\$}\texttt{x}\texttt{\textasciicircum}\texttt{\backslash overline\{m+n+k\}} \\ & \text{\texttt{\backslash quad}\texttt{x}\texttt{\textasciicircum}\texttt{\backslash underline\{n\}}\texttt{\$}\texttt{\$}} \end{aligned}$$

22.7. Греческие символы

Греческие символы вводятся следующими командами:

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

$$\Xi^2 = \zeta^2 + \theta^2 + \eta^2 \quad \text{\texttt{\$}\texttt{\$}\texttt{\backslash Xi}\texttt{\textasciicircum}2\texttt{=}\texttt{\backslash zeta}\texttt{\textasciicircum}2\texttt{+}\texttt{\backslash theta}\texttt{\textasciicircum}2\texttt{+}\texttt{\backslash eta}\texttt{\textasciicircum}2\texttt{\$}\texttt{\$}}$$

$$S = \pi R^2 \quad \text{\texttt{\$}\texttt{\$}\texttt{S}\texttt{=}\texttt{\backslash pi}\texttt{\textasciicircum}2\texttt{\$}\texttt{\$}}$$

22.8. Математические операторы

+	<code>\pm</code>	-	<code>\le</code>	=	<code>\ge</code>
±	<code>\mp</code>	<	<code>\gg</code>	>	<code>\equiv</code>
∓	<code>\ll</code>	≤	<code>\cdots</code>	≡	<code>\ldots</code>
≪	<code>\cdot</code>	≫	<code>\colon</code>	...	<code>\mid</code>
·	<code>\ne, \neq</code>	:	<code>\smile</code>		<code>\frown</code>
≠	<code>\circ</code>	∴	<code>\prime</code>	(	<code>\forall</code>
◊	<code>\aleph</code>	/	<code>\emptyset</code>	∃	<code>\exists</code>
ℵ	<code>\hbar</code>	∅	<code>\nabla</code>	¬	<code>\neg, \lnot</code>
ℏ	<code>\partial</code>	∇	<code>\surd</code>	⋮	<code>\flat</code>
∂	<code>\infty</code>	√	<code>\top</code>	⋈	<code>\natural</code>
∞	<code>\ell</code>	⊥	<code>\bot, \perp</code>	#	<code>\sharp</code>
ℓ	<code>\wp</code>	⊥	<code>\ , \parallel</code>	△	<code>\triangle</code>
℘	<code>\Re</code>	∥	<code>\angle</code>	\	<code>\backslash</code>
ℜ	<code>\Im</code>	∠	<code>\mid</code>	≐	<code>\doteq</code>
ℑ	<code>\approx</code>		<code>\times</code>	*	<code>\ast</code>
≈	<code>\odot</code>	×	<code>\diamond</code>	•	<code>\bullet</code>
⋈	<code>\star</code>	◇	<code>\cup</code>	∩	<code>\cap</code>
⋆	<code>\div</code>	∪	<code>\uplus</code>	⊓	<code>\sqcap</code>
÷	<code>\propto</code>	⊕	<code>\triangleright</code>	◁	<code>\triangleleft</code>
∝	<code>\wr</code>	▷	<code>\vee, \lor</code>	∧	<code>\wedge, \land</code>
ℳ	<code>\asymp</code>	∨	<code>\ominus</code>	⊗	<code>\otimes</code>
ℴ	<code>\oplus</code>	⊖	<code>\dagger</code>	‡	<code>\ddagger</code>
ℵ	<code>\oslash</code>	†	<code>\succ</code>	~	<code>\sim</code>
ℶ	<code>\prec</code>	⋈	<code>\succeq</code>	≈	<code>\simeq</code>
ℷ	<code>\preceq</code>	⋈	<code>\supset</code>	≡	<code>\amalg</code>
ℸ	<code>\subset</code>	⊃	<code>\supseteq</code>	≅	<code>\cong</code>
ℹ	<code>\subseteq</code>	⊇	<code>\sqsupseteq</code>	☞	<code>\bowtie</code>
℺	<code>\sqsubseteq</code>	⊈	<code>\ni, \owns</code>	∉	<code>\notin</code>
℻	<code>\in</code>	∋	<code>\dashv</code>	⊨	<code>\models</code>
ℼ	<code>\vdash</code>	⊣			

Чтобы сделать оператор отношения отрицательным, следует ввести перед ним команду `\not` (для двух конструкций, $\neq$ и $\notin$, имеются специальные команды `\ne` и `\notin`).

$x * y / z$	$x \times y \cdot z$	$a \approx b \sim c \simeq d$	$s \cong l$	<code>\$x*y/z\quad x\times y\cdot z\quad</code>	<code>a\approx b\sim c\simeq d\quad s\cong l\$</code>
$x \cup y \cap z$	$x \neq y$	$x \vee y \wedge z$	$x \notin D$	<code>\$\$\$x\cup y\cap z\quad x\neq y\quad</code>	<code>x\vee y\wedge z\quad x\notin D\$\$\$</code>
$S \subseteq \Sigma \iff S \in \mathcal{S}$				<code>\$\$\$S\subseteq\Sigma\iff S\in\{\cal S\}\$\$\$</code>	
$\angle\alpha$	$\triangle ABC$	$L \perp M$	$N \parallel L$	<code>\$\$\$ \angle\alpha\quad \triangle ABC\quad</code>	<code>L\perp M\quad N\parallel L\quad x\not\equiv y\$\$\$</code>

22.9. “Большие” математические операторы

Σ	<code>\sum</code>	$\int$	<code>\int</code>	$\oint$	<code>\oint</code>
$\prod$	<code>\prod</code>	$\coprod$	<code>\coprod</code>	$\amalg$	<code>\amalg</code>
$\bigvee$	<code>\bigvee</code>	$\bigodot$	<code>\bigodot</code>	$\biguplus$	<code>\biguplus</code>
$\bigcirc$	<code>\bigcirc</code>	$\bigtriangleup$	<code>\bigtriangleup</code>	$\bigtriangledown$	<code>\bigtriangledown</code>
$\bigcap$	<code>\bigcap</code>	$\bigcup$	<code>\bigcup</code>	$\bigvee$	<code>\bigvee</code>
$\bigwedge$	<code>\bigwedge</code>	$\bigoplus$	<code>\bigoplus</code>	$\bigotimes$	<code>\bigotimes</code>

Размер некоторых больших операторов — знака суммы, интеграла и других — зависит от того, выделена или нет в отдельную строку формула, содержащая их.

$\bigcup_n X_n \bigcap \bigcap_n Y_n$	$\prod_i n_i$	<code>\bigcup_n X_n \bigcap \bigcap_n Y_n</code>	<code>\prod_i n_i</code>
$M := \bigvee \{U : U \in \omega t\}$		<code>M := \bigvee \{U : U \in \omega t\}</code>	
$\bigcap_i C_i \bigcap \bigcap_j C_j = \bigcirc \text{ при } i = j$		<code>\bigcap_i C_i \bigcap \bigcap_j C_j = \bigcirc \text{ при } i = j</code>	
$\bigcup_n X_n \bigcap \bigcap_n Y_n$	$\prod_i n_i$	<code>\bigcup_n X_n \bigcap \bigcap_n Y_n</code>	<code>\prod_i n_i</code>
$M := \bigvee \{U : U \in \omega t\}$		<code>M := \bigvee \{U : U \in \omega t\}</code>	
$\bigcap_i C_i \bigcap \bigcap_j C_j = \bigcirc \text{ при } i = j$		<code>\bigcap_i C_i \bigcap \bigcap_j C_j = \bigcirc \text{ при } i = j</code>	

22.10. Стандартные математические функции

Стандартным математическими функциями здесь считаются те, которые по правилам полиграфии должны набираться в формулах не математическим курсивом, а прямым шрифтом — элементарные тригонометрические и алгебраические функции и т.д.

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

$$\begin{array}{ll}
 x' = x \cos \xi + y \sin \xi & \text{\texttt{\$x'=x\cos\xi+y\sin\xi\$}} \\
 x'' = -x \sin \xi + y \cos \xi & \text{\texttt{\$x''=-x\sin\xi+y\cos\xi\$}} \\
 \text{mod } Z & \text{\texttt{\$mod Z \quad \pmod Z \quad}} \\
 (\text{mod } Z) & \text{\texttt{\$arcsin\{abc\}\quad \ln R\$}} \\
 \arcsin abc & \text{\texttt{\$sin\{\phi\}}^2+\{\cos\{\xi\}\}^2=1\$} \\
 \ln R & \\
 \sin \phi^2 + \cos \xi^2 = 1 &
 \end{array}$$

22.11. Стрелки

$\rightarrow$	<code>\to (\rightarrow)</code>	$\leftarrow$	<code>\gets (\leftarrow)</code>
$\leftrightarrow$	<code>\leftrightharrow</code>	$\Rightarrow$	<code>\Rightarrow</code>
$\Leftarrow$	<code>\Leftarrow</code>	$\Leftrightarrow$	<code>\Leftrightharrow</code>
$\longrightarrow$	<code>\longrightarrow</code>	$\longleftarrow$	<code>\longleftarrow</code>
$\longleftrightarrow$	<code>\longleftrightharrow</code>	$\Longrightarrow$	<code>\Longrightarrow</code>
$\Longleftarrow$	<code>\Longleftarrow</code>	$\iff$	<code>\iff (\Longleftrightharrow)</code>
$\uparrow$	<code>\uparrow</code>	$\downarrow$	<code>\downarrow</code>
$\updownarrow$	<code>\updownarrow</code>	$\Uparrow$	<code>\Uparrow</code>
$\Downarrow$	<code>\Downarrow</code>	$\Updownarrow$	<code>\Updownarrow</code>
$\mapsto$	<code>\mapsto</code>	$\longmapsto$	<code>\longmapsto</code>
$\nearrow$	<code>\nearrow</code>	$\hookrightarrow$	<code>\hookrightarrow</code>
$\hookleftarrow$	<code>\hookleftarrow</code>	$\searrow$	<code>\searrow</code>
$\rightharpoonup$	<code>\rightharpoonup</code>	$\leftharpoonup$	<code>\leftharpoonup</code>
$\swarrow$	<code>\swarrow</code>	$\rightharpoonrightarrow$	<code>\rightharpoonrightarrow</code>
$\leftharpoonrightarrow$	<code>\leftharpoonrightarrow</code>	$\nwarrow$	<code>\nwarrow</code>
$\rightharpoonrightarrow$	<code>\rightharpoonrightarrow</code>		

$$\begin{array}{ll}
 R(n, t) = O(t^{n/2}), \text{ если } t \rightarrow 0^+ & \text{\texttt{\$R(n,t)=O(t^{\{n/2\}}),~если~t\to 0^+ \$}} \\
 L(k) \iff R(k) \text{ и } L(k) \iff R(k) \text{ эквивалентны.} & \text{\texttt{\$mbox\{L\}(k)\iff mbox\{R\}(k) \$ и}} \\
 & \text{\texttt{\$rm L\}(k)}} \\
 & \text{\texttt{\$Longleftrightharrow\{rm R\}(k) \$}} \\
 & \text{\texttt{эквивалентны.}}
 \end{array}$$

22.12. Пределы, суммы, интегралы

▷ `\lim_{<нижн. инд.>}{<аргумент>}`

Знак предела функции или последовательности вводится командой `\lim`. Выражение со стрелкой, определяющее предел аргумента функции или изменение номера члена последовательности, вводится как обычный нижний индекс.

Формула $\lim_{n \rightarrow \infty} \left(\frac{n+1}{n}\right)^n = e$ неудобна в строке. Лучше

$$\lim_{n \rightarrow \infty} \left(\frac{n+1}{n}\right)^n = e$$

$$u'(t) = \lim_{\Delta t \rightarrow 0} \frac{u(t + \Delta t) - u(t)}{\Delta t}$$

Формула $\lim_{n \rightarrow \infty} \left(\frac{n+1}{n}\right)^n = e$ неудобна в строке. Лучше

$$\lim_{n \rightarrow \infty} \left(\frac{n+1}{n}\right)^n = e$$

$\mathbf{u}'(t) = \lim_{\Delta t \rightarrow 0} \frac{\mathbf{u}(t + \Delta t) - \mathbf{u}(t)}{\Delta t}$

▷ $\sum_{\langle \text{нижн. инд.} \rangle}^{\langle \text{верх. инд.} \rangle} \langle \text{аргумент} \rangle$

Знак суммы вводится командой `\sum`. При вводе суммы в выражение в двух знаках доллара индексы суммирования будут напечатаны под и над знаком суммы. Если знак суммы имеется в выражении, заключенном в одинарные знаки доллара, то знак суммы будет несколько меньше, а индексы суммирования будут сдвинуты так, чтобы следовать знаку суммы, а не находиться над и под ним.

Формула $e^x = \sum_{m=0}^{\infty} \frac{x^m}{m!} = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots$ неудобна в строке. Лучше

$$e^x = \sum_{m=0}^{\infty} \frac{x^m}{m!} = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots$$

Формула

$e^x = \sum_{m=0}^{\infty} \frac{x^m}{m!} = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots$

неудобна в строке.

Лучше

$e^x = \sum_{m=0}^{\infty} \frac{x^m}{m!} = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots$

▷ $\int_{\langle \text{нижн. пред.} \rangle}^{\langle \text{верх. пред.} \rangle} \langle \text{выражение} \rangle$

Знак интегрирования вводится командой `\int`. Введение интегралов аналогично введению сумм. Пределы задаются как нижние и верхние индексы. Небольшой пробел между подинтегральной функцией и дифференциалом следует вводить командой `\, , .`

Формула $\int_a^b f(x) dx = F(b) - F(a)$ неудобна в строке. Лучше

$$\int_a^b f(x) dx = F(b) - F(a)$$

Формула $\int_a^b f(x) \, dx = F(b) - F(a)$ неудобна в строке. Лучше

$$\int_a^b f(x) \, dx = F(b) - F(a)$$

22.13. Радикалы, дроби, биномиальные коэффициенты

▷ `\sqrt [<степень>] {<аргумент>}`

Выдает знак корня над аргументом. При построении квадратного корня необязательный параметр `<степень>` может быть опущен.

L^AT_EX варьирует высоту корня в зависимости от высоты аргумента.

$\sqrt{x+y}$	$\sqrt{Z}$	<code>\$\$\sqrt{x+y}\qquad\sqrt{Z}\$\$</code>
$\sqrt[n+1]{x^n+y^n} + \sqrt{Z}$		<code>\$\$\sqrt[n+1]{x^n+y^n}+\sqrt{Z}\$\$</code>
$\sqrt[n+1]{x^n+y^n} + \sqrt{Z}$		<code>\$\$\sqrt{x+y}+\sqrt{Z}\qquad</code>
$\sqrt{x+y} + \sqrt{Z}$	$\sqrt[3]{x+y + \sqrt[n]{z}}$	<code>\sqrt[3]{x+y +\sqrt[n]{z}}\$\$</code>
$\cos(2\alpha) = \sqrt{1 - 2\sin^2(\alpha)}$		<code>\$\$\cos{(2\alpha)}=</code> <code>\sqrt{1-2\sin^2{(\alpha)}}\$\$</code>

▷ `\frac {<числитель> } {<знаменатель>}`

Создает дробь с заданными числителем и знаменателем.

$\frac{x+b+y/d}{x^2-y^2+a}$	$\frac{\frac{a+b}{c+d} + \frac{a+x}{a+d}}{\frac{x+a}{c+b}}$	<code>\$\$\frac{x+b+y/d}{x^2-y^2+a}\qquad</code> <code>\frac{\frac{a+b}{c+d}+\frac{a+x}{a+d}}{\frac{x+a}{c+b}}\$\$</code>
$x_0 + \frac{1}{x_1 + \frac{1}{x_2 + \frac{1}{x_3 + \frac{1}{x_4}}}}$		<code>\$\$x_0+\frac{1}{x_1+\frac{1}{x_2+\frac{1}{x_3+\frac{1}{x_4}}}}</code> <code>{x_2+\frac{1}{x_3+\frac{1}{x_4}}}}\$</code> <code>\bigskip</code> <code>\$\$x_0+\frac{1}{x_1+</code> <code>\frac{\displaystyle 1}{\displaystyle x_2+\frac{1}{\displaystyle x_3+\frac{1}{\displaystyle x_4+\frac{1}{x_5+\frac{1}{x_6}}}}}</code> <code>{x_6}}}}}}\$</code>

Если требуется записать одно выражение над другим без разделяющей черты, удобно использовать команду T_EX'a `\atop`, которую следует поместить в общие группирующие скобки с обоими этими выражениями.

`{<текст сверху> \atop <текст внизу>}`

Для записи биномиальных коэффициентов можно использовать T_EX'овскую команду `\choose`, аналогичную по формату команде `\atop`.

`{<символ сверху> \choose <символ внизу>}`

$$P_{r-j} = \begin{matrix} \binom{n+i}{n} & \text{если } r-j \text{ четное} \\ 0 & \text{если } r-j \text{ нечетное} \end{matrix}$$

$\$n+i \choose n\$$
 $\$P_{r-j}=\{r!(-1)^{(r-j)/2}\atop 0\}$
 $\{\mbox{если}~ r-j \mbox{~четное~}\atop \mbox{если}~r-j\mbox{~нечетное}\}\$$

22.14. “Большие” группирующие операторы и вертикальные стрелки

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

Эти операторы доступны в пяти размерах:

- нормальный размер;
- `\big`, `\bigl`, `\bigr` — чуть больше нормального размера;
- `\Big`, `\Bigl`, `\Bigr` — в 1.5 раза больше размера, даваемого операторами размера `\big`;
- `\bigg`, `\biggl`, `\biggr` — в 2.0 раза больше размера, даваемого операторами размера `\big`;
- `\Bigg`, `\Biggl`, `\Biggr` — в 2.5 раза больше размера, даваемого операторами размера `\big`.

Для левых частей группирующих операторов используются команды с символом ‘l’ в конце, для правых — команды с последним символом ‘r’. Для вертикальных линий и стрелок используются команды, указанные первыми.

$$\Bigg\uparrow () \{ \} \backslash | || \lceil \rceil$$

$\$ \Bigg\uparrow \sim \Bigl(\sim \Bigr) \sim$
 $\Biggl\{ \sim \Biggr\} \sim \Biggr\backslash$
 $\sim \big\vert \sim \big\Vert \sim$
 $\bigl\lceil \sim \bigr\rceil \$$

Приведенный здесь набор дает представление о том, как выглядят разделители всех пяти размеров.

() [] { } / \ { } [] [] < > | || ↑ ↓ ↗ ↘ ⇅ ⇆

() [] { } / \ { } [] [] < > | || ↑ ↓ ↗ ↘ ⇅ ⇆

() [] { } / \ { } [] [] < > | || ↑ ↓ ↗ ↘ ⇅ ⇆

() [] { } / \ { } [] [] < > | || ↑ ↓ ↗ ↘ ⇅ ⇆

() [] { } / \ { } [] [] < > | || ↑ ↓ ↗ ↘ ⇅ ⇆

Существует также возможность задать величины скобок в соответствии с вертикальным размером помещаемого в них выражения. Для этого перед открывающей скобкой следует поместить команду `\left`, а перед закрывающей — команду `\right`.

Вообще говоря, скобки должны присутствовать в математическом выражении парами. $\text{\LaTeX}$ проверяет наличие команды `\left` для каждой команды `\right` и наоборот.

Если требуется ввести лишь одну скобку, например, обобщающую, то соответствующую ей «невидимую» скобку следует указать надлежащей командой с параметром в виде точки.

$$\left(\sqrt{1 + \sqrt{1 + \sqrt{1 + \sqrt{1 + \sqrt{1 + x}}}}} \right)$$

```

 $\left(\sqrt{1+\sqrt{1+\sqrt{1+\sqrt{1+\sqrt{1+x}}}}}\right)$ 
 $\uparrow\uparrow k{\mathop{=}}^{\mathop{def}} 2^{\left.2^{\cdot^{\cdot^{\cdot^2}}}\right\}_k}$ 

```

$$2 \uparrow\uparrow k \stackrel{def}{=} 2^{\left.2^{\cdot^{\cdot^{\cdot^2}}}\right\}_k}$$

22.15. Матрицы

Построение матриц аналогично построению таблиц, но в математической моде. И формат самих командных скобок ввода массивов, и используемые в пределах этих скобок команды практически идентичны.

Командные скобки, формирующие массивы, в том числе матрицы:

```

▷ \begin{array}[<положение>]{<столб.1>...<столб.n>}
    <вход 11> & ... & <вход 1n> \\ [<величина>]
    .....
\end{array}

```

Имеют один обязательный параметр, описывающий число столбцов и способ выравнивания элементов в них.

Символ `&` является разделителем элементов в строке,

Команда `'\\'` является признаком конца строки. Эта команда может иметь необязательный параметр `<величина>`, увеличивающий расстояние между этой и следующей строкой на заданную величину.

Скобки имеют следующие параметры.

- `<положение>`

Указывает позицию по вертикали относительно текущей строки: выравнивание по верхнему ряду (`t`), выравнивание по нижнему ряду (`b`). По умолчанию бокс выравнивается по центру.

- `<столб.1>...<столб.n>`

Обязательный параметр, указывающий способ форматирования столбцов и наличие разделяющих столбцы вертикальных линий. Для каждого столбца должно быть задано одно из приведенных ниже значений:

`l` – выравнивание колонки по левому краю;

`r` – выравнивание колонки по правому краю;

`c` – выравнивание колонки относительно центра;

`|` – вертикальная линия на полную высоту и глубину бокса;

`@{text}` – вставка текста в каждый ряд таблицы. @-выражение подавляет обычный промежуток между столбцами, поэтому любой промежуток между вставленным текстом и смежными элементами должен быть включен в текст явно.

`p{wd}` – создает столбец, в котором каждый элемент печатается как абзац шириной `wd`. Команда `'\\'` не может появляться в подобном элементе за исключением следующих ситуаций: (i) внутри командных скобок, подобных `minipage`, `array`, или `tabular`; (ii) внутри явно выданных скобок `parbox` или в пределах текста, формируемого командами `\centering`, `\raggedright`, или `\raggedleft`. Последние команды могут появиться внутри групповых или командных скобок, которые используются в элементах `p`-столбца.

Ниже приведены команды, используемые в пределах командных скобок `array`.

```

▷ \cline{i-j}

```

Рисует горизонтальные линии через столбцы, номера которых указаны в обязательном параметре.

▷ `\hline`

Рисует горизонтальную линию по ширине массива. Может быть выдана до, после или между строками массива.

▷ `\multicolumn{<столбцы>}{<положение>}{<текст>}`

Используется для объединения нескольких элементов строки в один.

Все параметры являются обязательными.

Первый указывает число столбцов для объединения.

Второй указывает на выравнивание объединенного столбца относительно центра (c),
левого края (l);
правого края (r).

Третий параметр задает текст. Эта команда должна либо начинать строку, либо следовать сразу же за символом разделения элементов '&'.

▷ `\vline`

Рисует вертикальную линию на полную высоту и глубину строки. Для сдвига линии в пределах столбца можно использовать команду `\hfill`, которая может также использоваться в @-выражении.

$$\begin{pmatrix} \cos kl & \frac{1}{k} \sin kl & 0 \\ 0 & -k \sin kl & \cos kl \\ 0 & 0 & 1 \end{pmatrix}$$

```
$$\left(\begin{array}{ccc}
\cos{kl}&\frac{1}{k}\sin{kl}&0\\
0&-k\sin{kl}&\cos{kl}\\
0&0&1
\end{array}\right)$$
```

$$\mathcal{M} \leftarrow \begin{array}{c} \mathcal{O} \\ | \\ XY \\ | \\ \infty \end{array} \rightarrow \mathcal{N}$$

```
$$\begin{array}{ccccc}
&&\mathcal{O}&& \\
&&|&& \\
\mathcal{M}&\leftarrow&XY&\rightarrow&\mathcal{N} \\
&&|&& \\
&&\infty&&
\end{array}$$
```

$$|x| = \begin{cases} x & \text{при } x \geq 0; \\ -x & \text{при } x < 0. \end{cases}$$

```
$$|x|=\left\{\begin{array}{l}
x \quad \text{при } x \geq 0; \\
-x \quad \text{при } x < 0.
\end{array}\right. $$
```

22.16. Уравнения

Командные скобки построения уравнения или системы уравнений автоматически переводят L^AT_EX в математическую моду. Ввод знаков $\$ \dots \$$ или $\$ \$ \dots \$ \$$ не нужен.

▷ `\begin{equation}`
`<уравнение>`
`\end{equation}`

Командные скобки построения уравнения. Помещают уравнение в центре строки, и справа от него ставят его порядковый номер.

$x' = x \cos \beta + y \sin \beta$	(1)	<code>\begin{equation}</code> <code>x' = x\cos{\beta} + y\sin{\beta}</code>
$y' = -x \sin \beta + y \cos \beta$	(2)	<code>\end{equation} \begin{equation}</code> <code>y' = -x\sin{\beta} + y\cos{\beta}</code>
$x^2 + y^2 = x'^2 + y'^2$	(3)	<code>\end{equation} \begin{equation}</code> <code>x^2+y^2 = {x'}^2 + {y'}^2</code> <code>\end{equation}</code>

▷ `\begin{eqnarray}`
`<левая часть ур-ния 1> & <знак> & <правая часть ур-ния 1> \\\`
`.....`
`<левая часть ур-ния n> & <знак> & <правая часть ур-ния n>\nonumber\\`
`<уравнение n>`
`\end{eqnarray}`

Командные скобки построения системы уравнений или неравенств.

Вводят массив с тремя столбцами. Строки массива разделяются символами `\\`, элементы в строке — символом `&`.

Выравнивание делается автоматически по схеме `lcr`.

Номер уравнения ставится на каждой строке, для которой не выдана команда `\nonumber`.

▷ `\begin{eqnarray*}`
`<левая часть ур-ния 1> & <знак> & <правая часть ур-ния 1> \\\`
`.....`
`<левая часть ур-ния n> & <знак> & <правая часть ур-ния n> \\\`
`\end{eqnarray*}`

Если командные скобки `eqnarray` заданы в форме со звездочкой, то номера уравнений не проставляются.

▷ `\lefteqn{<текст>}`

Команда, используемая для смещения влево строк командных скобок `eqnarray` и `eqnarray*`. Полезна при печати длинных формул. `<Текст>` будет первой строкой, выданной стандартным для этих скобок образом. Последующие строки будут смещены влево относительно начала первой строки. Строки разделяются символами `\\`.

$ \begin{array}{rcl} X_1 + \cdots + X_n & = & l \\ Y_1 + \cdots + Y_n & = & m \\ Z_1 + \cdots + Z_n & = & n \end{array} $	(1)	<pre> \setcounter{equation}{0} \begin{eqnarray} X_1+\cdots+X_n & = & l\backslash\mathrm{nonumber}\\ Y_1+\cdots+Y_n & = & m \qquad \qquad \backslash\backslash \\ Z_1+\cdots+Z_n & = & n \backslash\mathrm{nonumber} \end{eqnarray} </pre>
$ \begin{array}{rcl} S + T + L & = & a + b + a^2 + b^2 + c^2 + \\ & & M * (e + f) \end{array} $		<pre> \begin{eqnarray*} S+T+L & = & a+b+a^2+b^2+c^2+\backslash\backslash \\ & & M*(e+f) \quad \backslash\backslash \\ \end{eqnarray*} \begin{eqnarray*} \lefteqn{S+T+L=} \quad \backslash\backslash \\ & & a+b+a^2+b^2+c^2+M*(e+f) \quad \backslash\backslash \\ \end{eqnarray*} </pre>
$ \begin{array}{rcl} S + T + L = & & \\ a + b + a^2 + b^2 + c^2 + M * (e + f) \end{array} $		

$\text{\LaTeX}$ задает сквозную нумерацию математических выражений. Чтобы номер уравнения в этом примере был 1, счетчик уравнений `equation` установлен в 0 командой `\setcounter`. В противном случае этот номер был бы равен 4, так как один пример с уравнениями уже был дан на странице 82.

23. Определение новых команд

Команды данного раздела позволяют вводить новые команды и командные скобки, а также переопределять команды и командные скобки.

- ▷ `\newcommand{\<имя>}[<кол-во арг.>]{<определение>}`
- ▷ `\renewcommand{\<имя>}[<кол-во арг.>]{<определение>}`

Команды определяют (или переопределяют) команду и имеют следующие параметры:

- <имя>** имя команды, которому должен предшествовать символ `\`. Оно не должно быть определено ранее для `\newcommand`; для `\renewcommand` имя должно уже существовать;
- <кол-во арг.>** целое число от 1 до 9, определяющее количество параметров новой команды. По умолчанию команда не имеет параметров;
- <определение>** набор команд, исполняемых при вызове данной команды; формальный параметр `#n` заменяется при этом текстом n-ого аргумента. Если вновь определяемая команда содержит символы в математической моде и предназначена для использования в любой моде, то математические символы в ней следует заключать в бокс.

- ▷ `\newenvironment{<имя>}[<кол-во арг.>]{<начало>}{<конец>}`
- ▷ `\renewenvironment{<имя>}[<кол-во арг.>]{<начало>}{<конец>}`

Эти команды определяют или переопределяют командные скобки и имеют следующие параметры:

- <имя>** имя командных скобок. Для `\newenvironment` не должно быть определено ранее и команда `\<имя>` также не должна быть определена. Для `\renewenvironment` имя должно уже существовать;
- <кол-во арг.>** целое число от 1 до 9, определяющее количество параметров новых командных скобок. По умолчанию командные скобки не имеют параметров;
- <начало>** список команд, которые будут исполняться в месте вызова начала командных скобок `\begin{<имя>}`; формальный параметр `#n` заменяется при этом текстом n-ого аргумента;
- <конец>** команды, которые будут исполняться в месте вызова конца командных скобок `\end{<имя>}`. Они не должны содержать никаких параметров.

- ▷ `\newtheorem{<имя>}{<заголовок>}[<счетчик>]`
- ▷ `\newtheorem{<имя>}[<аналог>]{<заголовок>}`

Создает командные скобки определения теоремы. Имеет параметры:

- <имя>** имя командных скобок, которое не должно совпадать ни с одним именем существующих командных скобок или счетчика;

<заголовок>	текст, который печатается в начале командных скобок, непосредственно перед номером теоремы;
<счетчик>	имя уже определенного счетчика, обычно это номер раздела. Обеспечивает задание номера для новой теоремы в пределах этого раздела;
<аналог>	имя уже определенных командных скобок, указывающее, что новые теоремы должны нумероваться в одной последовательности с теоремами типа <аналог>.

Командные скобки ввода теорем являются нумеруемыми. Для них имеется специальный счетчик, позволяющий задавать соответствующей скобке порядковый номер.

Построение вот таких отрезков

┌ ┌ ┌

```
\newcommand{\rv}{\rule{.4pt}{4pt}} % Вертикальная черточка
\newcommand{\rvb}{\rv\hrulefill\rv}% Линия в пределах черточек
\makebox[5mm]{\rvb}\qqquad \makebox[10mm]{\rvb}\qqquad \makebox[15mm]{\rvb}
```

В данном примере теоремы типа «аксиома» нумеруются в пределах секции, а теоремы типа «правило» — в пределах полной работы.

Аксиома 23.1 *Прямая есть кратчайшее расстояние между двумя точками.*

Правило 1 $A + B = B + A$

Правило 2 $A + (B + C) = A + B + C$

```
\newtheorem{axiom}{Аксиома}[section]
\newtheorem{algebr}{Правило}
\newtheorem{alg}[algebr]{Правило}
\begin{axiom} Прямая есть кратчайшее расстояние между двумя точками.\end{axiom}
\begin{algebr}  $A + B = B + A$ \end{algebr}
\begin{alg}  $A + (B + C) = A + B + C$ \end{alg}
```

23.1. Построение макрокоманд средствами Т_ЕX’a

Т_ЕX имеет более мощные и гибкие средства для введения макрокоманд, ведь он первичен по отношению к Л_АT_ЕX’у. Макрокоманды, описанные в файле Plain Т_ЕX и в Т_ЕXbook’e, являются *развиваемыми*, они очень эффективны и остроумно используют многие аспекты языка примитивов. На их основе собственно и написан Л_АT_ЕX.

В настоящем руководстве мы приводим очень краткую и поверхностную информацию о компоновке макрокоманд. Однако в практическом отношении этот раздел важен

— пользователь сможет начать строить удобный для него набор макрокоманд, и изучение надлежащих разделов книги Д.Кнута не будет для него прыжком в неизвестность.

Процедура написания макрокоманд на $\TeX$ 'е достаточно сложная. Отладка макрокоманд также чревата проблемами, т.к. $\TeX$ в большой степени является интерпретируемым языком: ошибка в макрокоманде не может быть обнаружена, пока макрокоманда не используется (если только это не простая синтаксическая ошибка типа потери группирующей скобки).

Макрокоманда есть совокупность сгруппированных команд и фрагментов текста, объединяемых общим названием.

Введение в исходный текст этого названия вызовет исполнение всех команд данной макрокоманды и вставку всех определенных в ней фрагментов. Если внутренние команды включают параметры, макрокоманда строится так, чтобы при вызове ее можно было задать все требуемые параметры.

Команда определения макрокоманды содержит имя, названия параметров, если они нужны, и тело макрокоманды, причем названия параметров в командах тела макрокоманды должны совпадать с теми, что заданы при имени макрокоманды. Поскольку с помощью названий в этих командах лишь фиксируются положения параметров, принято говорить, что определение макрокоманды может включать формальные параметры.

Команда вызова макрокоманды есть введение ее имени и фактических значений параметров. При исполнении макрокоманды эти значения будут подставлены вместо соответствующих формальных параметров, и, таким образом, будет выполнена именно требуемая конкретная последовательность команд.

Простейшая форма определения макрокоманд в $\TeX$ 'е следующая:

```
\def<Имя>#1#2...{.....#1.....#2.....}
```

<Имя> есть название, присвоенное макрокоманде, уникальное в наборе макрокоманд.

Символ # и последующее число задают формальное название параметра (их может не быть или быть не более 9).

Группирующие скобки содержат фрагмент, подставляемый в документ в месте указания в нем данной макрокоманды.

Пример макрокоманды без параметров дает макрокоманда $\TeX$. Мы вводим лишь символы $\backslash\TeX$, тогда как сама команда определения этой макрокоманды есть

```
\def\TeX{T\kern-.1667em\lower.5ex\hbox{E}\kern-.125emX}
```

Приведем еще несколько примеров.

```
% Макрокоманды, используемые при наборе нашей книги
\def\cmm#1{{\tt $\backslash$#1}}      % Ввод названия команды
\def\sty#1{{\tt #1}}                  % Ввод названия стилевого файла

% Макрокоманды формата Plain Tex
\def\quad{\hskip1em\relax}            % Определения команд ввода
\def\qqquad{\hskip2em\relax}          % различных пробелов
```

<code>\def\line{\hbox to\hsize}</code>	% Команда определения строки
<code>\centerline#1{\line{\hss#1\hss}}</code>	% Центрирование текста на строке

Замечание. Присутствующая в примерах команда `\relax` есть признак того, что предшествующая команда окончена, и дальнейший текст не следует воспринимать как параметр. Предосторожность, не лишняя в определениях макрокоманд, за которыми в исходных текстах может следовать совершенно произвольный фрагмент.

24. Команды работы со счетчиками и размерными величинами

Здесь обсуждаются команды, редко используемые при обычном наборе текста, но важные для пользователей, готовых писать собственные макрокоманды.

24.1. Счетчики

Для каждого набора однородных нумеруемых объектов, например, рубрик или командных скобок ввода уравнений, в $\text{\LaTeX}$ выделяется счетчик. Имя счетчика совпадает с именем командной скобки или команды, влияющей на этот счетчик. Стандартные стили документов $\text{\LaTeX}$ используют следующие счетчики:

part	paragraph	figure	enumi
chapter	subparagraph	table	enumii
section	page	footnote	enumiii
subsection	equation	mpfootnote	enumiv
subsubsection			

Цель введения счетчиков первых трех столбцов, следующая из их названий, очевидна (счетчик `mpfootnote` отведен под нумерацию сносок на мини-странице).

Счетчики четвертого столбца используются при перечислениях. Списки перечислений могут быть вложенными, максимальная глубина вложения — четыре.

Для операций со счетчиками $\text{\LaTeX}$ располагает следующими командами.

- ▷ `\addtocounter {<счетчик>} {<число>}`
Добавляет к счетчику число, определенное вторым параметром, значение которого может быть отрицательным.
- ▷ `\alph{<счетчик>}`
Приводит к печати строчной буквы алфавита, соответствующей значению счетчика. Команда `\Alph` приводит к печати прописных букв алфавита.
- ▷ `\arabic{<счетчик>}`
Приводит к печати значения счетчика арабскими цифрами.
- ▷ `\fnsymbol{<счетчик>}`
Приводит к печати одного из девяти специальных символов, используемых для нумерации сносок. Номер символа определен значением счетчика, указанного в данной команде.
- ▷ `\newcounter{<счетчик>} [<связанный счетчик>]`
Создает новый счетчик, имя которого не должно быть определено ранее. Указание связанного счетчика в необязательном параметре приведет к установке нового счетчика в нуль всякий раз, когда изменяется значение этого связанного счетчика.
- ▷ `\usecounter{<счетчик>}`
Позволяет использовать значение указанного в ней счетчика, например, для нумерации пунктов в командных скобках формирования списков.

▷ `\roman{<счетчик>}`

Приводит к печати значения счетчика римскими цифрами. Команда `\roman` вызывает строчные цифры римской нумерации. Команда `\Roman` вызывает прописные цифры римской нумерации.

▷ `\setcounter{<счетчик>}{<число>}`

Устанавливает значение счетчика, указанное во втором обязательном параметре.

▷ `\value{<счетчик>}`

Извлекает значение названного счетчика. Может использоваться там, где $\text{\LaTeX}$ ожидает целое число, например, как второй параметр команды `\setcounter` или `\addtocounter`. Полезна для арифметических действий со счетчиками.

Напечатаем номер текущей секции различными способами:

24 Ч ч XXIV xxiv

```
\arabic{section} \quad \Alph{section} \quad \alph{section} \quad
\Roman{section} \quad \roman{section}
```

В стандартных базовых стилях счетчики команд рубрикации для подсекций и подподсекций введены так, чтобы номера младших рубрик начинались с 1 в каждой новой старшей рубрике:

```
\newcounter{subsection}[section]
\newcounter{subsubsection}[subsection]
```

В стандартных базовых стилях для сносок, уравнений и систем уравнений принята сквозная нумерация. Для того, чтобы в каждой главе или секции номера сносок, уравнений и систем уравнений начинались с 1, следует в начале каждой главы (или секции) выдавать команды

```
\setcounter{footnote}{0} \setcounter{equation}{0} \setcounter{eqnarray}{0}
```

Определение отступа, равного абзацному, умноженному на значение указанного счетчика:

```
\hspace{\value{<счетчик>}\parindent}
```

24.2. Длины

В командах $\text{\LaTeX}$ 'а длины могут быть жесткими или эластичными (т.е. обладать способностью сжиматься и растягиваться). Существуют команды и соответствующие им специальные единицы измерения, задающие длине бесконечную эластичность: `fil`, `fill`, `filll`. Единицы `fil` рекомендуется использовать в пределах боксов, `fill` — в пределах строки или страницы, `filll` — бесконечность, с которой надо обращаться осторожно.

Жесткие единицы измерения приведены в таблице 2 на стр. 22.

С длинами работают следующие команды.

- ▷ `\newlength{\<имя>}`
Определяет команду длины со значением ‘0pt’. Команда `<\имя>` не должна быть определена ранее.
- ▷ `\setlength {\<имя>} {\<длина>}`
Используется для установки значения данной команды длины. Параметр `<длина>` может быть отрицательным.
- ▷ `\addtolength {\<имя>} {\<длина>}`
Изменяет значение команды длины на указанную величину. Значение `<длина>` может быть отрицательным.
- ▷ `\settowidth {\<имя>} {\<текст>}`
Устанавливает значение команды длины, равное ширине параметра `<текст>`.

Пусть мы хотим сделать фрагмент, ширина которого на 2 см больше ширины его заголовка.

Подготовка материалов, завершающих руководство

Последняя стадия подготовки руководства по программному обеспечению включает планирование приложений, подготовку раздела «Сообщения об ошибках», сводов правил синтаксиса в программах и процедур ввода данных, а также построение глоссария и различных указателей. Готовый текст должен пройти проверку на читабельность.

```
% Вводим новую длину и задаем ее значение
\newlength{\minip}
\settowidth{\minip}{\large\sl Подготовка материалов, завершающих руководство}
\addtolength{\minip}{2cm}
% Собственно построение фрагмента текста
\begin{center}
{\large \sl Подготовка материалов, завершающих руководство}\\
\medskip
\begin{minipage}{\minip}
Последняя стадия подготовки руководства по программному обеспечению включает
планирование приложений, подготовку раздела <Сообщения об ошибках>, сводов
правил синтаксиса в программах и процедур ввода данных, а также построение
глоссария и различных указателей. Готовый текст должен пройти проверку на
читабельность.
\end{minipage} \end{center}
```

25. Параметры стиля

Здесь перечисляются внутренние команды $\text{\LaTeX}$ 'а (предназначенные для построения файлов *описания стиля*), которые пользователь может переопределять в собственном документе.

Эти команды, которые далее будут называться *параметрами стиля*, задаются в формате команд $\text{\TeX}$ 'а.

- Символ признака команды $\backslash$ сопровождается мнемоникой, которой присваивается аргумент.
- Аргумент задается числом, размерной величиной или последовательностью символов, не содержащей цифр, а также командой, значение которой уже определено. Знак равенства между мнемоникой и аргументом не является обязательным, его обычно включают для наглядности.

Некоторые параметры стиля из файла `art12.sty`.

```
\textwidth 390pt    % Ширина текста на странице
\headheight 12pt    % Высота бокса для колонтитула
\parindent 1.5em     % Величина абзацного отступа
\arraycolsep 5pt     % Полуширина промежутка между столбцами таблицы или массива
```

Параметры стиля построения страницы и ее фрагментов

$\backslash\text{oddsidemargin}$	На право-определенной странице расстояние от левого края листа до левого поля текста, без учета стандартного отступа в один дюйм,
$\backslash\text{evensidemargin}$	На лево-определенной странице расстояние от правого края листа до правого поля текста, без учета стандартного отступа в один дюйм,
$\backslash\text{marginparwidth}$	Ширина заметок на полях.
$\backslash\text{marginparsep}$	Горизонтальное расстояние между текстом и заметкой на полях.
$\backslash\text{topmargin}$	Расстояние от верхнего края листа до первой строки на странице, без учета стандартного отступа в один дюйм.
$\backslash\text{headheight}$	Высота бокса для колонтитула.
$\backslash\text{headsep}$	Вертикальное расстояние между заголовком и телом страницы.
$\backslash\text{textheight}$	Высота тела страницы.
$\backslash\text{textwidth}$	Ширина текста на странице.
$\backslash\text{topskip}$	Минимальное расстояние между началом тела страницы и базовой линией первой строки текста. Аналог $\backslash\text{baselineskip}$ для первой строки страницы.
$\backslash\text{footheight}$	Высота бокса для сноски.
$\backslash\text{footskip}$	Расстояние между базовыми линиями последней строки страницы и первой сноски.

<code>\baselineskip</code>	Расстояние между основаниями соседних строк.
<code>\parindent</code>	Величина абзацного отступа.
<code>\parskip</code>	Дополнительный вертикальный промежуток между абзацами.

Параметры стиля для построения математических выражений

<code>\mathindent</code>	Для отмены центрирования математических формул, печатаемых отдельной строкой, используется корректирующий стиль <code>fleqn</code> . Этот параметр задает отступ от левого поля для формул, печатаемых отдельной строкой.
<code>\abovedisplayskip</code>	Дополнительное пространство, оставляемое над длинной формулой. (В стиле <code>fleqn</code> используется команда <code>\topsep</code> .)
<code>\belowdisplayskip</code>	Дополнительное пространство, оставляемое под длинной формулой. (В стиле <code>fleqn</code> используется команда <code>\topsep</code> .)
<code>\abovedisplayshortskip</code>	Дополнительное пространство, оставляемое над короткой формулой. (В стиле <code>fleqn</code> используется команда <code>\topsep</code> .)
<code>\belowdisplayshortskip</code>	Дополнительное пространство, оставляемое под короткой формулой. (В стиле <code>fleqn</code> используется команда <code>\topsep</code> .)
<code>\arraycolsep</code>	Полуширина промежутка между столбцами массива.
<code>\doublerulesep</code>	Ширина промежутка между линиями, создаваемыми при выдаче в командных скобках построения массива двух последовательных команд <code>\hline</code> или <code>\vline</code> .
<code>\arrayrulewidth</code>	Ширина линий, создаваемых командами <code>\hline</code> и <code>\vline</code> .
<code>\arraystretch</code>	Контроль промежутков между рядами массива.

Параметры стиля для построения таблиц

<code>\arraycolsep</code>	Полуширина промежутка между столбцами.
<code>\doublerulesep</code>	Ширина промежутка между линиями, создаваемыми при выдаче двух последовательных команд <code>\hline</code> или <code>\vline</code> .
<code>\arraycolsep</code>	Полуширина промежутка между столбцами.
<code>\doublerulesep</code>	Ширина промежутка между линиями, создаваемыми при выдаче двух последовательных команд <code>\hline</code> или <code>\vline</code> .
<code>\arrayrulewidth</code>	Ширина линий, создаваемых командами <code>\hline</code> и <code>\vline</code> .
<code>\arraystretch</code>	Контроль промежутков между рядами.

26. Диагностика ошибок

Исходный текст документа может находиться в нескольких файлах, вызываемых в корневом файле командами `\input` или `\include`.

В процессе обработки документа $\TeX$ информирует пользователя о том, какой файл он обрабатывает: при открытии нового файла на экран выводится открывающая круглая скобка, за которой следует полное имя файла; далее выдаются номера страниц, сформированных $\TeX$ 'ом из информации, содержащейся в данном файле; наконец, по окончании обработки файла на экран выводится закрывающая скобка. Скобки, обрамляющие имя корневого файла, будут содержать вложенные скобки с полными именами всех используемых файлов и списками созданных ими страниц. Таким образом пользователь имеет возможность следить, какой именно файл обрабатывается в данный момент.

Сообщения об ошибках, выдаваемые $\TeX$ 'ом, сопровождаются указанием номера строки в виде `l.<номер>`.

Пусть выдача на экран имеет следующий вид:

```
(C:\TEX\DOC\RUSSIAN\INTRO.TEX [1] [2] [3] (C:\TEX\DOC\RUSSIAN\APP.TEX [4]
[5]) [6]
! Undefined control sequence.
1.321 \began{picture}
?
```

Корневой файл документа называется `INTRO.TEX`. После формирования первых трех страниц начинается обработка файла `APP.TEX`. После формирования следующих двух страниц обработка файла `APP.TEX` заканчивается, и $\TeX$ возвращается к обработке файла `INTRO.TEX`. Сформировав еще одну страницу, $\TeX$ обнаруживает ошибку на строке 321 текущего файла.

При обнаружении ошибки возможны следующие действия:

- нажав клавишу `E` или `X`, прекратить обработку и исправить ошибку с помощью редактора;
- нажав клавишу `RETURN`, дать $\TeX$ 'у указание продолжить работу до обнаружения следующей ошибки;
- нажав клавишу с символом `R`, дать $\TeX$ 'у указание продолжить работу до конца, игнорируя эту и последующие ошибки.

В двух последних случаях $\TeX$ попытается исправить ошибку (что дает положительный результат лишь в простейших случаях пропуска группирующей скобки или одного из разделителей математической моды) и продолжить обработку документа.

Наиболее часто встречающиеся ошибки, распознаваемые $\TeX$ 'ом и $\LaTeX$ 'ом:

- несоответствие открывающих и закрывающих группирующих скобок или скобок, выделяющих аргумент команды;
- пропущен один из разделителей математической моды;
- команда использована в недопустимой для нее моде.
- пропущена команда начала или конца документа;

- пропущен аргумент команды;

26.1. Сообщения об ошибках, выдаваемые L^AT_EX’ом

Здесь приводится полный список сообщений, выдаваемых L^AT_EX’ом при нахождении ошибок. Сообщения приведены в алфавитном порядке.

- **Bad \line or \vector argument**
Первый аргумент команд `\line` или `\vector`, используемых внутри командных скобок `picture`, имеет неправильные компоненты.
- **Bad math environment delimiter**
Пропущен один из разделителей, обрамляющих выражения в математической моде: `$`, `$$`, `\[`, `\]`, `\(` или `\)`.
- **Bad use of **
Команда `\\` используется между абзацами, где она не имеет смысла.
- **\begin{...} ended by \end{...}**
L^AT_EX нашел только конец или только начало командных скобок. Возможная причина — опечатка в имени, а также пропуск или лишняя вставка `\begin` или `\end` командных скобок.
- **Can be used only in preamble**
Команда L^AT_EX’а, которая может появляться только в преамбуле документа, использована после команды `\begin{document}`. К командам преамбулы относятся: `\documentstyle`, `\nofile`, `\includeonly`, `\makeindex` и `\makeglossary`. Та же диагностика появляется при наличии лишней команды `\begin{document}`.
- **Command name ... already used**
Попытка использовать одну из команд определения новой команды или командной скобки `\newlist`, `\newtheorem`, `\newcommand`, или `\newlength` с именем, которое было использовано ранее. Если вы хотите переопределить уже определенную команду, вместо `\newcommand` нужно использовать `\renewcommand`.
- **Counter too large**
L^AT_EX пытается сделать перечисление, отмечая пункты буквами, и при этом число пунктов превышает 26 (количество букв в английском алфавите). Возможна также ошибка в работе с другими счетчиками.
- **Environment ... undefined**
Используются командные скобки с неизвестным именем.
- **Float(s) lost**
Командные скобки плавающих элементов (`figure`, `table`) или команда `\marginpar` помещены внутри бокса. Если в тексте много плавающих элементов, L^AT_EX, возможно, обнаружит эту ошибку, далеко уйдя от содержащей ее команды, и найти соответствующее место в тексте будет нелегко. Один из таких элементов будет потерян, но необязательно тот, который ошибочно помещен в бокс.

- **Illegal character in array arg**
В аргументе командных скобок `array` или `tabular` или во втором аргументе команды `\multicolumn`, используемой в этих командных скобках, найден запрещенный символ.
- **Missing `\begin{document}`**
Пропущена команда `\begin{document}`, либо в преамбуле содержится текст или команды, формирующие текст.
- **Missing p-arg in preamble**
В аргументе командных скобок `array` или `tabular`, или во втором аргументе команды `\multicolumn`, используемой внутри этих командных скобок, найден символ ‘p’ без следующего за ним выражения.
- **Missing @-exp in preamble**
В аргументе командных скобок `array` или `tabular`, или во втором аргументе команды `\multicolumn`, используемой внутри этих командных скобок, найден символ ‘@’ без следующего за ним @-выражения.
- **No such counter**
Указание несуществующего счетчика в одной из команд работающей со счетчиками, например `\setcounter` или `\addtocounter`. Если ошибка появляется при обработке файла с расширением `.aux`, это может означать, что определение счетчика командой `\newcounter` находится вне преамбулы (т.е. после `\begin{document}`).
- **Not in outer par mode**
Командные скобки плавающих элементов (`figure`, `table`) или команда `\marginpar` используются в математической моде либо в боксе. Плавающие элементы можно использовать только во внешней абзацной моде.
- **`\pushtabs` and `\poptabs` don’t match**
В пределах командных скобок `tabbing` обнаружена команда `\poptabs` без предшествующей команды `\pushtabs`; или в момент обработки команды `\end{tabbing}` оказалось, что одна или несколько команд `\pushtabs` лишние.
- **Something’s wrong--perhaps a missing `\item`**
Скорее всего, пропущена команда `\item` в командных скобках, создающих список. Такое сообщение выдается, если пропущен аргумент в командных скобках `thebibliography`.
- **Tab overflow**
Превышен предел для числа позиций табулятора. Увеличение этого предела может быть достигнуто только модификацией L^AT_EX’a.
- **There’s no line here to end**
Команды `\newline` или ‘\’ используются между абзацами, где они не имеют смысла. Для получения дополнительных промежутков между абзацами следует использовать команды `\vspace` или `\vskip`.
- **This is a LaTeX bug**
Не повезло. Это следствие ошибки в самом L^AT_EX’е. В таком случае рекомендуется показать документ эксперту.

- **Too deeply nested**

В документе более шести вложенных списков или более четырех вложенных списков с автоматическим перечислением. (Командные скобки `quote` и `verse` также рассматриваются как списки.)

- **Too many unprocessed floats**

Л^AT_EX'у не хватило памяти для плавающих элементов, создаваемых командными скобками `figure`, `table` или командой `\marginpar`. Ошибка возникает либо при попытке создать слишком много таких объектов на одной странице, либо когда в командных скобках `figure` или `table` заданы параметры размещения, превышающие возможности одной страницы.

- **Undefined tab position**

Попытка перемещения табулятура на неопределенную позицию при использовании одной из следующих команд: `\>`, `\+`, `\-` или `\<`.

- **\< in mid line**

Команда `'\<'` появилась в середине строки в командных скобках `tabular`. Эта команда может находиться только в начале строки.

26.2. Сообщения об ошибках, выдаваемые TeX'ом

При обработке файла Л^AT_EX'ом ряд ошибок диагностируется самим процессором T_EX. Здесь приводится список сообщений об ошибках, выдаваемых T_EX'ом.

- **Counter too large**

Сноски нумеруются буквами или специальными символами, число которых, разумеется, ограничено. Эта ошибка может возникнуть, например, при использовании слишком большого числа команд `\thanks`.

- **Double subscript**

В математической формуле обнаружена неоднозначная комбинация для получения двойного нижнего индекса.

- **Double superscript**

В математической формуле обнаружена неоднозначная комбинация для получения двойного верхнего индекса.

- **Extra alignment tab has been changed to \cr**

Количество символов `'&'` в строке массива или таблицы превышает заданное количество столбцов.

- **Extra }, or forgotten \$**

Группирующие скобки не соответствуют друг другу, или пропущен (либо задан лишний) ограничитель выражения в математической моде.

- **Font ... not loaded. Not enough room left**

Документ использует больше шрифтов, чем позволяет T_EX. Можно попробовать обработать документ по частям.

-
- I can't find file '...'

TeX не может найти указанный входной файл. Это может быть результатом ошибки при вводе имени файла при запуске TeX'a или указанием несуществующего файла в команде `\input` или `\documentstyle`.

Если файл не найден, TeX выдает приглашение:

```
Please type another input file name:
```

и ждет ввода надлежащего имени файла.

- Illegal unit of measure (pt inserted)

При задании длины указана безразмерная величина.

- Illegal parameter number in definition of ...

При использовании команд `\newcommand` или `\renewcommand` сделана ссылка на аргумент, номер которого больше числа аргументов, определенных в этой команде. Если вместо многоточия в сообщении указана команда `\@gtempa`, это может означать, что в аргумент команды `\label` помещен символ '#'.
• Misplaced alignment tab character &

Использование специального символа '&' в обычном тексте, а не в таблице.

- Missing control sequence inserted

В определении новых команд с помощью `\newcommand`, `\renewcommand` или `\newlength` есть такое, в котором первый аргумент не является правильно заданным именем команды.

- Missing number, treated as zero

Не задано числовое значение параметра (размерного или безразмерного). Скорее всего пропущен обязательный аргумент команды, или квадратные скобки в тексте помещены так, что были приняты за начало необязательного аргумента.

- Missing { inserted

- Missing } inserted

Пропущена одна из группирующих скобок.

- Missing \$ inserted

Ряд команд TeX'a могут быть использованы только в математической моде. Встречая такие команды вне математической моды, TeX выдает данное сообщение об ошибке и при этом автоматически переключается в математическую моду.

- Not a letter

Ошибка в аргументе команды `\hyphenation`.

- Runaway argument?

- Paragraph ended before ... was complete

Зарегистрирован признак конца абзаца (команда `\par` или пустая строка) в аргументе команды, в котором не может быть нескольких абзацев. Причиной ошибки может быть также отсутствие закрывающей скобки в конце аргумента или разделителя в команде `\verb`.

- **TeX capacity exceeded, sorry [...]**

TeX'у не хватило памяти, и он прекратил работу. В квадратных скобках будет указан параметр, определяющий возможные причины ошибки, именно:

buffer size В аргументах команд рубрикации или команд `\caption`, `\addcontentsline`, или `\addtocontents` задан слишком длинный фрагмент текста. Сообщение может быть выдано при работе команды `\end{document}`, если подобная ошибка имела место при исполнении команд `\tableofcontents`, `\listoffigures` или `\listoftables`.

exception dictionary TeX'у задано слишком много исключений для переноса слов командами `\hyphenation`. Удалите из этих команд редко встречающиеся в тексте слова, используя для указания в них места переноса команду `\-`.

hash size Определено слишком много команд и(или) меток для перекрестных ссылок.

main memory size Создана настолько сложная страница, что TeX не может хранить всю информацию, необходимую для ее формирования. Для того чтобы выявить причину, по которой переполняется память, рекомендуется вставить команду `\clearpage` перед местом возникновения ошибки. Затем, возможно, придется изменить исходный файл в месте сбоя, упростив построение страниц (например, перенеся подальше друг от друга рисунки и таблицы).

pool size Возможно, вы используете слишком много перекрестных ссылок и(или) определяете слишком много новых команд. Точнее, у ваших меток и команд слишком большая суммарная длина в символах. Попробуйте сделать имена ваших команд и меток короче.

Такая диагностика может также появиться, если пропущена закрывающая скобка, отделяющая аргументы таких команд, как `\setcounter`, `\newenvironment` или `\newtheorem`.

save size Команды, командные скобки и группы определений имеют слишком глубокую вложенность. Чаще всего это следствие заикливания команд, когда команда вызывает сама себя, возможно через несколько уровней.

- **Text line contains an invalid character**

Строка содержит запрещенные символы. Скорее всего текст имеет неправильную кодировку.

- **Undefined control sequence**

Использована неопределенная команда. Наиболее вероятная причина — наличие ошибки в имени команды или имени макропакета. Возможно не задан файл описания стиля, содержащий определение данной команды.

- **Use of ... doesn't match its definition**

Если `'...'` — команда L^AT_EX'a, скорее всего это одна из команд, используемых в пределах командных скобок `picture`, аргумент которой задан в неверном формате.

Если `'...'` — `\@array`, неправильно задано @-выражение в командной скобке `array`.

- **You can't use 'macro parameter character #' in ... mode**

Использование специального символа `'#'` в обычном тексте. Повидимому, требовалось ввести `'\#'`.

26.3. Предупреждающие сообщения L^AT_EX'a

В отличие от процедуры выдачи сообщения об ошибке, при выдаче предупреждающих сообщений L^AT_EX не приостанавливает работу. Все L^AT_EX'овские предупреждения начинаются с текста: 'LaTeX Warning'. Ниже приводятся в алфавитном порядке предупреждения, выдаваемые L^AT_EX'ом.

- Citation '...' on page ... undefined
Метка цитируемой работы в команде \cite, используемой при построении списка литературы, не определена командой \bibitem.
- Label '...' multiply defined
Несколько команд \label или \bibitem имеют один и тот же аргумент.
- Label(s) may have changed. Rerun to get cross-references right
Числа, выводимые командами \ref, \pageref или \cite могут оказаться неверными. Рекомендуется выполнить еще один прогон через L^AT_EX.
- Marginpar on page ... moved
Сноска на полях, создаваемая командой \marginpar, будет сдвинута вниз во избежание вывода ее поверх предыдущей сноски. Таким образом, данная сноска не окажется напротив той строки, на которой встретила команду \marginpar.
- Oval too small
Команде \oval, используемой в командных скобках picture, поручено нарисовать настолько маленький овал, что выполнить это не представляется возможным. Овал поэтому будет несколько больше заданного.
- Reference '...' on page ... undefined
Аргумент команды \ref или \pageref, используемых при установлении перекрестных ссылок, не был определен командой \label.

26.4. Предупреждающие сообщения T_EX'a

T_EX свои предупреждения никак не предваряет и работу не приостанавливает. Ниже приводятся предупреждения, выдаваемые T_EX'ом.

- Overfull \hbox ...
Текст не помещается в горизонтальный бокс. Зачастую такая ситуация возникает, когда T_EX не может удачно разбить абзац на строки.
Исправить ситуацию можно несколькими способами:
 - задать дополнительные места разрешенных переносов слов;
 - поставить принудительный или предпочтительный разрыв строк в некоторых местах;
 - если таких предупреждений возникает слишком много, можно увеличить ширину страницы.

В этом сообщении выводится также информация о том, насколько переполнен бокс. Если это переполнение не очень велико, сообщение можно игнорировать.

- **Overfull \vbox ...**

TeX не может найти удачного места для перехода на новую страницу, поэтому на страницу выводится текст, превышающий размер страницы. Рекомендуется использовать соответствующие команды для указания наиболее удачных мест для перехода на следующую страницу.

- **Underfull \hbox**

Горизонтальный бокс почти или вовсе не заполнен. Такое сообщение может быть вызвано, например, использованием команды `\linebreak` в строке, заполненной менее чем наполовину.

- **Underfull \vbox**

TeX не может найти удачного места для перехода на новую страницу, и создает страницу с недостаточным количеством текста на ней. В таком случае рекомендуется использовать соответствующие команды для указания наиболее удачных мест для перехода на следующую страницу.

Термины и определения

Настоящий список содержит глоссарий терминов и определений, которые используются в обеих частях данного руководства.

- **Визуализатор**
Специальный драйвер для просмотра сформатированного документа на экране дисплея.
- **Гарнитура**
Набор шрифтов, сохраняющих стилевые признаки основного рисунка, но отличающиеся по начертаниям и размерам. Изменение рисунка шрифта в пределах одной гарнитуры могут дать начертания, различающиеся
 - по наклону — прямое, наклонное вправо или влево;
 - по плотности — узкое, нормальное, широкое;
 - по насыщенности — светлое, жирное.
- Гарнитура (как и кегль) есть классическая полиграфическая характеристика шрифта.
- **Глиф**
Графическое представление символа. Например, **a**, **а** и **Ⓐ** есть три различных глифа для одного и того же символа. термин образован сокращением английского слова GLYptograPHy, означающего "резьба на драгоценных камнях".
- **Драйверы**
Программы, обеспечивающие вывод создаваемых **TeX**'ом и **LaTeX**'ом не зависящих от типа внешнего устройства файлов на конкретное устройство вывода.
- **Кегль**
Размер шрифта в типографских пунктах, включающий высоту буквы (очка) и заплечики (свободное пространство над и под очком). Кегль относится к классическим полиграфическим характеристикам шрифта.
- **Лигатуры**
Типографское представление двух или трех символов в виде одного "иероглифа", облегчающее их зрительное восприятие. Например, символы **"ffi"** или **"---**" **TeX** выдаст как **"ffi"** и **"—"**.
- **Макрокоманда**
Произвольная команда, составленная из примитивов и(или) других макрокоманд. Макрокоманда может иметь параметры.
- **METAFONT**
Программа-спутник процессора текстов **TeX**, используемая для генерации новых шрифтов или существующих шрифтов с новым увеличением. Программа **METAFONT** читает файл определения шрифта, фактические параметры устройства, увеличение и т.д., чтобы создать *два* выходных файла на каждый входной файл определения шрифта. Входной файл должен быть файлом типа **.mf**, а два выходных файла есть соответственно **tmf**- и **gf**-файлы. Если вы хотите сделать шрифт пяти различных увеличений, вы должны пять раз запустить на счет програм-

му METAFONT с почти одинаковым mf-файлом. Или если вы желаете создать шрифт для трех различных устройств (матричное печатающее устройство, лазерное устройство печати, дисплей) вы опять-таки должны запустить программу METAFONT три раза с соответствующим определением устройства. *Все три раз будет выдан один и тот же ttf-файл.* Однако gf-файлы будут *различными*.

- Метрический файл шрифта

Аппаратно-независимый файл, содержащий набор чисел, определяющий для каждого глифа необходимое ему пространство на странице: размер бокса, заключающего в себя глиф и некоторые граничные условия, например относительные расстояния между соседними глифами. Тип файла есть ttf.

- New Font Selection Scheme — NFSS

Новая схема выбора шрифтов (New Font Selection Scheme — NFSS) введена в последних версиях L^AT_EX 2.09, L^AT_EX 3. В NFSS реализованы следующие важные моменты:

- Команды изменения характеристик шрифта полностью независимы и могут использоваться в любом порядке.
- Команда загрузки любого шрифта может быть выдана в любом месте документа.

- PAW

Система интерактивного анализа данных, интегрирующая в себе различные инструменты управления и представления данных. Графические пакеты включают возможность создания файлов в формате POSTSCRIPT.

- Пиксель

Независимый элемент растрового представления графической информации, который способно напечатать устройство. Разрешение устройства вывода определяется в точках(пикселях) на дюйм, Dot per inch, DPI, отдельно по горизонтали и вертикали. Для механических устройств типично разрешение 240DPI по горизонтали и 180DPI по вертикали. Для лазерных устройств типичное разрешение есть 300DPI. Обычно указывается разрешение только по горизонтали.

- Примитивы

Около трех сотен встроенных в программу T_EX команд, неразложимых на более простые функциональные компоненты.

- POSTSCRIPT

Язык программирования для устройств печати (разработка фирмы Adobe), используемый на многих лазерных устройствах.

- Разрешение (характеристика шрифта)

Разрешение как характеристика шрифта определяется разрешением устройства вывода, для которого этот шрифт предназначен. Измеряется в точках(пикселях) на дюйм и обычно указывается в имени растрового файла в качестве его расширения.

- Растривание символа

METAFONT имеет свою внутреннюю графическую решетку или растр, состоящий из прямоугольных ячеек, внутри которых можно поместить только один пиксель. Растривание символа есть создание пиксельной картинке его путем ука-

- зания METAFONT'у, какие пиксели раstra должны быть "черными", а какие "белыми". По существу это есть преобразование формы в двоичные комбинации.
- Растровый шрифтовой файл
Файл, содержащий битовое представление шрифта для конкретного устройства вывода. Очевидно, что растровое представление любого глифа в шрифте явно зависит от устройства печати и увеличения, и каждый шрифт должен храниться в отдельном файле.
 - Стил ь документа
Совокупность макрокоманд, задающих для L^AT_EX'а тип создаваемого документа. Стил ь бывает базовым и корректирующим. Базовый стил ь определяет стандарт документа, распознаваемый L^AT_EX'ом: книга, статья, письмо, доклад, препринт. Корректирующий стил ь вносит в базовый поправки, дополняющие его. Например, указание корректирующего стили я "russian" при любом базовом стили я приведет к тому, что переносы в словах русских фрагментов документа будут сделаны по правилам русского языка.
 - Типографский пункт
Основная единица размерности, используемая в типографском деле. Один типографский пункт равен 0,3759 мм.
 - Типографский размер шрифта (design size)
Размер, выбранный при проектировании шрифта данного начертания. Измеряется в типографских пунктах. Программа METAFONT, может создавать посредством механического увеличения аналогичные шрифты других, отличных от типографского, размеров.
 - Формат
Заданный набор макрокоманд, откомпилированных и записанных в файл типа FMT, использование которого позволяет избежать траты времени на повторную компиляцию.
 - Шрифт (фонт)
Набор глифов, соответствующих набору символов.

Алфавитный указатель

<code>\</code> , 10	<code>\arraystretch</code> , 92
<code>\'</code> , 49, 50	<code>art10</code> (корр. стиль), 14
<code>\+</code> , 49, 96	<code>art11</code> (корр. стиль), 14
<code>\-</code> , 49, 50, 61, 96, 98	<code>art12</code> (корр. стиль), 14
<code>\<</code> , 49, 96	<code>art14</code> (корр. стиль), 22
<code>\=</code> , 49, 50	<code>article</code> (базовый стиль), 14
<code>\></code> , 49, 96	<code>\ast</code> , 73
<code>\#</code> , 98	<code>\asymp</code> , 73
<code>#</code> , 97, 98	<code>\atop</code> , 77
<code>&</code> , 51, 52, 80, 81, 97	<code>\author</code> , 65
<code>\\</code> , 24, 25, 49, 51, 58, 61, 80, 82, 94, 95	<code>\backslash</code> , 73, 78
<code>*</code> , 61	<code>\bar</code> , 71
<code>\'</code> , 50	<code>\baselineskip</code> , 91, 92
<code>10pt</code> (корр. стиль), 14, 15	<code>\begin</code> , 94
<code>11pt</code> (корр. стиль), 14	<code>{array}</code> , 51, 80, 95, 98
<code>12pt</code> (корр. стиль), 14	<code>{center}</code> , 24
<code>\a'</code> , 50	<code>{description}</code> , 30
<code>\a=</code> , 50	<code>{displaymath}</code> , 68
<code>\a'</code> , 50	<code>{document}</code> , 17
<code>\abovedisplayshortskip</code> , 92	<code>{enumerate}</code> , 30
<code>\abovedisplayskip</code> , 92	<code>{eqnarray}</code> , 82
<code>\acute</code> , 71	<code>{eqnarray*}</code> , 82
<code>\addcontentsline</code> , 35, 98	<code>{equation}</code> , 82
<code>\addtocontents</code> , 35, 98	<code>{figure}</code> , 33, 35, 94–96
<code>\addtocounter</code> , 88, 89, 95	<code>{flushleft}</code> , 25
<code>\addtolength</code> , 90	<code>{flushright}</code> , 25
<code>\addvspace</code> , 27	<code>{itemize}</code> , 30
<code>\aleph</code> , 73	<code>{math}</code> , 68
<code>\Alph</code> , 88	<code>{minipage}</code> , 40, 51, 80
<code>\alph</code> , 88	<code>{multicols}</code> , 64
<code>\alpha</code> , 72	<code>{parbox}</code> , 24, 80
<code>\amalg</code> , 73	<code>{picture}</code> , 56, 57, 94, 98
<code>\and</code> , 65	<code>{quote}</code> , 28
<code>\angle</code> , 73	<code>{quotation}</code> , 28
<code>\appendix</code> , 23	<code>{quote}</code> , 28, 96
<code>\approx</code> , 73	<code>{supertabular}</code> , 54
<code>\arabic</code> , 88	<code>{tabbing}</code> , 49, 50, 95
<code>\arccos</code> , 74	<code>{table}</code> , 24, 33, 35, 94–96
<code>\arcsin</code> , 74	<code>{tabular}</code> , 51, 80, 95
<code>\arctan</code> , 74	<code>{thebibliography}</code> , 37, 95
<code>\arg</code> , 74	<code>{titlepage}</code> , 66
<code>array</code> (ком. скобки), 51, 80, 95, 98	<code>{verbatim}</code> , 29
<code>\arraycolsep</code> , 92	<code>{verbatim*}</code> , 29
<code>\arrayrulewidth</code> , 92	<code>{verse}</code> , 28, 96

<code>\belowdisplayskip</code> , 92	<code>\cosh</code> , 74
<code>\belowdisplayskip</code> , 92	<code>\cot</code> , 74
<code>\beta</code> , 72	<code>\coth</code> , 74
<code>\bf</code> , 21	<code>\csc</code> , 74
<code>\bibitem</code> , 37, 99	<code>\cup</code> , 73
<code>\bigcap</code> , 74	<code>\dagger</code> , 73
<code>\bigcirc</code> , 74	<code>\dashbox</code> , 56
<code>\bigcup</code> , 74	<code>\dashv</code> , 73
<code>\bigodot</code> , 74	<code>\date</code> , 65
<code>\bigoplus</code> , 74	<code>\ddagger</code> , 73
<code>\bigotimes</code> , 74	<code>\ddot</code> , 71
<code>\bigskip</code> , 27	<code>\ddots</code> , 70
<code>\bigtriangledown</code> , 74	<code>\deg</code> , 74
<code>\bigtriangleup</code> , 74	<code>\Delta</code> , 72
<code>\biguplus</code> , 74	<code>\delta</code> , 72
<code>\bigvee</code> , 74	<code>description</code> (ком. скобки), 30
<code>\bigwedge</code> , 74	<code>\det</code> , 74
<code>\bmod</code> , 74	<code>\diamond</code> , 73
<code>book</code> (базовый стиль), 14	<code>\dim</code> , 74
<code>\bot</code> , 73	<code>displaymath</code> (ком. скобки), 68
<code>\bottomcaption</code> , 53	<code>\displaystyle</code> , 69
<code>\bowtie</code> , 73	<code>\div</code> , 73
<code>\breve</code> , 71	<code>document</code> (ком. скобки), 17
<code>\bullet</code> , 73	<code>\documentstyle</code> , 16–18, 94, 97
<code>\cal</code> , 69	<code>\dot</code> , 71
<code>\cap</code> , 73	<code>\doteq</code> , 73
<code>\caption</code> , 33, 34, 98	<code>\dotfill</code> , 45
<code>\cdot</code> , 73	<code>\doublerulesep</code> , 92
<code>\cdots</code> , 70, 73	<code>\Downarrow</code> , 75, 78
<code>center</code> (ком. скобки), 24	<code>\downarrow</code> , 75, 78
<code>\centering</code> , 24, 51, 80	<code>\downbracefill</code> , 46
<code>\chapter</code> , 23	<code>\ell</code> , 73
<code>chapter</code> (счетчик), 88	<code>\emptyset</code> , 73
<code>\check</code> , 71	<code>\end</code> , 94, 98
<code>\chi</code> , 72	<code>english</code> (корр. стиль), 18
<code>\choose</code> , 77	<code>\enspace</code> , 20
<code>\circ</code> , 73	<code>enumerate</code> (ком. скобки), 30
<code>\circle</code> , 56	<code>enumi</code> (счетчик), 88
<code>\circle*</code> , 56	<code>enumii</code> (счетчик), 88
<code>\cite</code> , 37, 38	<code>enumiii</code> (счетчик), 88
<code>\cleardoublepage</code> , 61	<code>enumiv</code> (счетчик), 88
<code>\clearpage</code> , 61, 98	<code>\epsilon</code> , 72
<code>\cline</code> , 51, 80	<code>eqnarray</code> (ком. скобки), 82
<code>\colon</code> , 73	<code>eqnarray*</code> (ком. скобки), 82
<code>\cong</code> , 73	<code>equation</code> (ком. скобки), 82
<code>\coprod</code> , 74	<code>equation</code> (счетчик), 88
<code>\cos</code> , 74	

<code>\equiv</code> , 73	<code>\hookleftarrow</code> , 75
<code>\eta</code> , 72	<code>\hookrightarrow</code> , 75
<code>\evensidemargin</code> , 91	<code>\hrule</code> , 32
<code>\exists</code> , 73	<code>\hrulefill</code> , 45
<code>\exp</code> , 74	<code>\hskip</code> , 26
<code>\extracolsep</code> , 51	<code>\hspace</code> , 26
<code>\fbox</code> , 45	<code>\hspace*</code> , 26, 51
<code>figure</code> (ком. скобки), 33, 35, 94–96	<code>\Huge</code> , 22
<code>figure</code> (счетчик), 88	<code>\huge</code> , 22
<code>\flat</code> , 73	<code>\hyphenation</code> , 61, 97, 98
<code>fleqn</code> (корр. стиль), 92	<code>\iff</code> , 75
<code>\flushbottom</code> , 64	<code>\Im</code> , 73
<code>flushleft</code> (ком. скобки), 25	<code>\imath</code> , 71
<code>flushright</code> (ком. скобки), 25	<code>\in</code> , 73
<code>\fnsymbol</code> , 88	<code>\include</code> , 60, 93
<code>\footheight</code> , 91	<code>\includeonly</code> , 60, 94
<code>\footnote</code> , 41	<code>\indent</code> , 23
<code>footnote</code> (счетчик), 88	<code>\inf</code> , 74
<code>\footnotemark</code> , 41, 42	<code>\infty</code> , 73
<code>\footnotesize</code> , 22	<code>\input</code> , 60, 93, 97
<code>\footnotetext</code> , 41, 42	<code>\int</code> , 74, 76
<code>\footskip</code> , 91	<code>\iota</code> , 72
<code>\forall</code> , 73	<code>\it</code> , 21
<code>\frac</code> , 77	<code>italian</code> (корр. стиль), 18
<code>\frame</code> , 57	<code>\item</code> , 30, 95
<code>\framebox</code> , 45, 57	<code>itemize</code> (ком. скобки), 30
<code>français</code> (корр. стиль), 18	<code>\jmath</code> , 71
<code>\frown</code> , 73	<code>\kappa</code> , 72
<code>\Gamma</code> , 72	<code>\ker</code> , 74
<code>\gamma</code> , 72	<code>\kill</code> , 49
<code>\gcd</code> , 74	<code>\label</code> , 34, 39, 97, 99
<code>\ge</code> , 73	<code>\Lambda</code> , 72
<code>germanb</code> (корр. стиль), 18	<code>\lambda</code> , 72
<code>\gets</code> , 75	<code>\land</code> , 73
<code>\gg</code> , 73	<code>\angle</code> , 78
<code>\grave</code> , 71	<code>\LARGE</code> , 22
<code>\hat</code> , 71	<code>\Large</code> , 22
<code>\hbar</code> , 73	<code>\large</code> , 22
<code>\hbox</code> , 46	<code>\lbrace</code> , 78
<code>\headheight</code> , 91	<code>\lceil</code> , 78
<code>\headsep</code> , 91	<code>\ldots</code> , 70, 73
<code>here</code> (корр. стиль), 34	<code>\le</code> , 73
<code>\hfill</code> , 26, 52, 81	<code>\left</code> , 79
<code>\hline</code> , 52, 81, 92	<code>\Leftarrow</code> , 75
<code>\hom</code> , 74	<code>\leftarrow</code> , 75

`\lefteqn`, 82
`\leftharpoondown`, 75
`\leftharpoonup`, 75
`\Leftrightarrow`, 75
`\leftrightharpoonup`, 75
letter (базовый стиль), 14
`\lfloor`, 78
`\lg`, 74
`\lim`, 74, 75
`\liminf`, 74
`\limsup`, 74
`\line`, 57, 94
`\linebreak`, 61, 100
`\linethickness`, 57
`\listoffigures`, 35, 98
`\listoftables`, 35, 98
`\ll`, 73
`\ln`, 74
`\lnot`, 73
`\log`, 74
`\Longleftarrow`, 75
`\longleftarrow`, 75
`\Longleftrightarrow`, 75
`\longleftrightarrow`, 75
`\longmapsto`, 75
`\Longrightarrow`, 75
`\longrightarrow`, 75
`\lor`, 73
LR мода, 44

`\makebox`, 45, 57
`\makeglossary`, 94
`\makeindex`, 94
`\maketitle`, 65
`\mapsto`, 75
`\marginpar`, 42, 94–96, 99
`\marginparsep`, 91
`\marginparwidth`, 91
`\markboth`, 63, 64
`\markright`, 63, 64
`math` (ком. скобки), 68
`\mathindent`, 92
`\mathop`, 70, 71
`\max`, 74
`\mbox`, 45
`\medskip`, 27
`\mid`, 73
`\min`, 74

`minipage` (ком. скобки), 40, 51, 80
`\models`, 73
`\mp`, 73
`mpfootnote` (счетчик), 88
`\mu`, 72
`multicol` (корр. стиль), 64
`multicols` (ком. скобки), 64
`\multicolumn`, 52, 81, 95
`\multiput`, 58

`\nabla`, 73
`\natural`, 73
`\ne`, 73
`\nearrow`, 75
`\neg`, 73
`\neq`, 73
`\newcommand`, 67, 84, 94, 97
`\newcounter`, 88, 95
`\newenvironment`, 84, 98
`\newlength`, 90, 94, 97
`\newline`, 61, 95
`\newlist`, 94
`\newpage`, 35, 61
`\newsavebox`, 47
`\newtheorem`, 84, 94, 98
`\ni`, 73
`\nofile`, 94
`\nofiles`, 36
`\noindent`, 24
`\nolinebreak`, 62
`\nonumber`, 82
`\nopagebreak`, 62
`\normalmagrinpar`, 42
`\normalsize`, 22
`\not`, 73
`\notin`, 73
`\nu`, 72
`\numbersign`, 21
`\narrow`, 75

`\o`, 72
`\oddsidemargin`, 91
`\odot`, 73
`\oint`, 74
`\Omega`, 72
`\omega`, 72
`\ominus`, 73
`\onecolumn`, 63

`\oplus`, 73
`\oslash`, 73
`\otimes`, 73
`\oval`, 57, 99
`\overbrace`, 71
`\overline`, 71
`\owns`, 73

`page` (счетчик), 88
`\pagebreak`, 62
`\pagenumbering`, 64
`\pageref`, 39, 99
`\pagestyle`, 63
`\par`, 23, 97
`\paragraph`, 23
`paragraph` (счетчик), 88
`\parallel`, 73
`\parbox`, 45, 51
`parbox` (ком. скобки), 24, 80
`\parindent`, 92
`\parskip`, 92
`\part`, 23
`part` (счетчик), 88
`\partial`, 73
`\perp`, 73
`\Phi`, 72
`\phi`, 72
`\Pi`, 72
`\pi`, 72
`picture` (ком. скобки), 56, 57, 94, 98
`Plain Tex`, 2, 85
`\pm`, 73
`\pmod`, 74
`\poptabs`, 50, 95
`\prec`, 73
`\preceq`, 73
`\prime`, 73
`\prod`, 74
`\propto`, 73
`\Psi`, 72
`\psi`, 72
`\pushtabs`, 50, 95
`\put`, 56

`\qqquad`, 20, 69
`\quad`, 20, 69
`quote` (ком. скобки), 28
`quote` (ком. скобки), 28, 96

`\raggedbottom`, 64
`\raggedleft`, 25, 51, 80
`\raggedright`, 25
`\raggedright`, 51, 80
`\raisebox`, 45
`\rangle`, 78
`\rbrace`, 78
`\rceil`, 78
`\Re`, 73
`\ref`, 39, 99
`\renewcommand`, 84, 94, 97
`\renewenvironment`, 84
`report` (базовый стиль), 14
`\reversemagrinpar`, 42
`\reversemarginpar`, 42
`\rfloor`, 78
`\rho`, 72
`\right`, 79
`\rightarrow`, 75
`\rightarrow`, 75
`\rightharpoondown`, 75
`\rightharpoonup`, 75
`\rightleftharpoons`, 75
`\rm`, 21
`\Roman`, 89
`\roman`, 89
`\rule`, 32
`russian` (корр. стиль), 18, 21

`\savebox`, 47
`\sc`, 21
`\scriptscriptstyle`, 69
`\scriptsize`, 22
`\scriptstyle`, 69
`\searrow`, 75
`\sec`, 74
`\section`, 23
`section` (счетчик), 88
`\selectlanguage`, 18
`\setcounter`, 89, 95, 98
`\setlength`, 56, 90
`\settowidth`, 90
`\sf`, 21
`\sharp`, 73
`\shortstack`, 58
`\Sigma`, 72
`\sigma`, 72
`\sim`, 73

<code>\simeq</code> , 73	<code>\textstyle</code> , 68
<code>\sin</code> , 74	<code>\textwidth</code> , 91
<code>\sinh</code> , 74	<code>\thanks</code> , 65, 96
<code>\sl</code> , 21	<code>thebibliography</code> (ком. скобки), 37, 95
<code>\small</code> , 22	<code>\Theta</code> , 72
<code>\smallskip</code> , 26	<code>\theta</code> , 72
<code>\smile</code> , 73	<code>\thinspace</code> , 20
<code>\space</code> , 67	<code>\thispagestyle</code> , 63
<code>spanish</code> (корр. стиль), 18	<code>\tilde</code> , 71
<code>\sqcap</code> , 73	<code>\times</code> , 73
<code>\sqrt</code> , 77	<code>\tiny</code> , 22
<code>\sqsubsepeq</code> , 73	<code>\title</code> , 65
<code>\sqsupsepeq</code> , 73	<code>titlepage</code> (ком. скобки), 66
<code>\stackrel</code> , 70	<code>\to</code> , 75
<code>\star</code> , 73	<code>\today</code> , 66
<code>\subparagraph</code> , 23	<code>\top</code> , 73
<code>subparagraph</code> (счетчик), 88	<code>\topcaption</code> , 53
<code>\subsection</code> , 23	<code>\topmargin</code> , 91
<code>subsection</code> (счетчик), 88	<code>\topsep</code> , 92
<code>\subset</code> , 73	<code>\topskip</code> , 91
<code>\subsepeq</code> , 73	<code>\triangleright</code> , 73
<code>\subsubsection</code> , 23	<code>\triangle</code> , 73
<code>subsubsection</code> (счетчик), 88	<code>\triangleleft</code> , 73
<code>\succ</code> , 73	<code>\tt</code> , 21
<code>\succeq</code> , 73	<code>\twocolumn</code> , 63
<code>\sum</code> , 74, 76	<code>\typein</code> , 67
<code>\sup</code> , 74	<code>\typeout</code> , 67
<code>\supertab</code> , 53	<code>\underbrace</code> , 71
<code>supertabular</code> (ком. скобки), 54	<code>\underline</code> , 72
<code>\supset</code> , 73	<code>\Uparrow</code> , 75, 78
<code>\supsepeq</code> , 73	<code>\uparrow</code> , 75, 78
<code>\surd</code> , 73	<code>\upbracefill</code> , 46
<code>\swarrow</code> , 75	<code>\Updownarrow</code> , 75, 78
<code>tabbing</code> (ком. скобки), 49, 50, 95	<code>\updownarrow</code> , 75, 78
<code>table</code> (ком. скобки), 24, 33, 94–96	<code>\uplus</code> , 73
<code>table</code> (счетчик), 88	<code>\Upsilon</code> , 72
<code>\tablecaption</code> , 53	<code>\upsilon</code> , 72
<code>\tablefirsthead</code> , 53	<code>\usebox</code> , 48
<code>\tablehead</code> , 53	<code>\usecounter</code> , 88
<code>\tablelasttail</code> , 53	<code>\usepackage</code> , 17
<code>\tableofcontents</code> , 35, 98	<code>\value</code> , 89
<code>\tabletail</code> , 53	<code>\varepsilon</code> , 72
<code>tabular</code> (ком. скобки), 51, 80, 95	<code>\varphi</code> , 72
<code>\tan</code> , 74	<code>\varpi</code> , 72
<code>\tanh</code> , 74	<code>\varrho</code> , 72
<code>\tau</code> , 72	<code>\varsigma</code> , 72
<code>\textheight</code> , 91	

—
\vartheta, 72
\vbox, 47
\vdash, 73
\vdots, 70
\vec, 71
\vector, 58, 94
\vee, 73
\verb, 29, 97
\verb*, 29
verbatim (ком. скобки), 29
verbatim* (ком. скобки), 29
verse (ком. скобки), 28, 96
\Vert, 74, 78
\vert, 78
\vfill, 27
\vline, 52, 81, 92
\vrule, 32
\vskip, 27, 95
\vspace, 27, 95
\vspace*, 27

\wedge, 73
\widehat, 72
\widetilde, 72
\wp, 73
\wr, 73

\Xi, 72
\xi, 72

\zeta, 72

Список литературы

- [1] Knuth D.E.: *The T_EXBook*. Addison-Wesley, Reading, Massachusetts, 1984.
- [2] Lamport L.: *L^AT_EX: A Document Preparation System*. Addison-Wesley, Reading, Massachusetts, 1986.
- [3] v.Bechtosheim Stephan, Old Oak Drive West Lafayette, USA. *T_EX in Practice*, 1989.
- [4] Sauver J.E.St., Oregon Eugene, USA: *Using T_EX on the VAX to Typeset Documents: a Primer*, 1990.
- [5] Глонти Н.Л., Грицаенко И.А. и др. *T_EX в ИФВЭ 5. Шрифты и работа с ними*. Препринт ИФВЭ 92–127, Протвино, 1992.
- [6] Глонти Н.Л., Грицаенко И.А. и др. *Многоязычный L^AT_EX*. Препринт ИФВЭ 94–57, Протвино, 1994.
- [7] Goossens M., Mittelbach F. and Samarin A.: *The L^AT_EX Companion*, Addison-Wesley, Reading, Massachusetts, 1994.
- [8] Diller A.: *L^AT_EX Line by Line*, John Wiley & Sons, Chichester, West Sussex, 1993.
- [9] Львовский С.М.: *Набор и верстка в пакете L^AT_EX*, Космоинформ, Москва, 1994.

Содержание

Предисловие	1
1. Общая характеристика $\text{\LaTeX}$ 'а	4
1.1. Что такое стиль документа	4
1.2. Основные возможности $\text{\LaTeX}$ 'а	4
1.3. Совместимость $\text{\LaTeX}$ 'а с другими форматирующими компонентами системы	5
2. Схема получения документа с помощью $\text{\LaTeX}$ 'а	6
3. Форматы команд $\text{\LaTeX}$ 'а	9
3.1. Единицы измерения	9
3.2. Управляющие символы	10
3.3. Команды	10
3.4. Командные скобки	11
3.5. Формат команд $\text{\TeX}$ 'а	12
4. Некоторые сведения о стилях	14
5. Команды задания стиля, начала и конца документа	16
5.1. Задание стиля в $\text{\LaTeX}$ 2.09	16
5.2. Задание стиля в $\text{\LaTeX}$ /2 ϵ и $\text{\LaTeX}$ 3	16
5.3. Указание начала и конца обрабатываемого текста	17
6. Выбор языка документа	18
7. Ввод простого текста	19
7.1. Символы, вводимые «как есть»	19
7.2. Пробелы	19
7.3. Символы, вводимые командами	20
7.4. Команды выбора шрифтов	21
7.4.1. Задание типа шрифта	21
7.4.2. Задание размера шрифта	22
7.5. Команды рубрикации	23
7.6. Построение абзаца	23
7.7. Центрирование фрагмента текста	24
7.8. Сдвиг фрагмента текста влево или вправо	25
7.9. Ввод горизонтальных промежутков	26
7.10. Ввод вертикальных промежутков	26
7.11. Цитирование	28
7.12. Буквальная передача текста	29
7.13. Составление списков	30
8. Горизонтальные и вертикальные линии	32

9. Плавающие элементы: рисунки и таблицы	33
10. Оглавление, списки рисунков и таблиц	35
11. Ввод библиографии	37
11.1. Построение списка литературы в тексте документа	37
11.2. Расстановка ссылок на литературу	38
12. Перекрестные ссылки	39
13. Построение мини-страницы	40
14. Сноски и заметки на полях	41
15. Боксы	44
15.1. Что такое боксы	44
15.2. Команды работы с боксами	44
15.3. Хранение текста в боксе	47
16. Таблицы	49
16.1. Выравнивание текста по шаблону	49
16.2. Построение произвольных таблиц	51
16.3. Как построить таблицу, длина которой превышает размер страницы . .	53
17. Построение диаграмм	56
18. Ввод текста из нескольких файлов	60
19. Разбиение на строки и страницы	61
20. Стили страниц, титульная информация	63
21. Вывод сообщений на терминал	67
22. Математические выражения	68
22.1. О размерах и стиле символов в формулах	68
22.2. Пробелы в математической моде	69
22.3. Построение многоточий различного типа	70
22.4. Верхние и нижние индексы	70
22.5. Символы со значками	71
22.6. Надстрочные и подстрочные знаки для группы символов	71
22.7. Греческие символы	72
22.8. Математические операторы	73
22.9. “Большие” математические операторы	74
22.10. Стандартные математические функции	74
22.11. Стрелки	75
22.12. Пределы, суммы, интегралы	75

22.13.Радикалы, дроби, биномиальные коэффициенты	77
22.14. “Большие” группирующие операторы и вертикальные стрелки	78
22.15.Матрицы	79
22.16.Уравнения	81
23. Определение новых команд	84
23.1. Построение макрокоманд средствами $\text{\TeX}$ ’а	85
24. Команды работы со счетчиками и размерными величинами	88
24.1. Счетчики	88
24.2. Длины	89
25. Параметры стиля	91
26. Диагностика ошибок	93
26.1. Сообщения об ошибках, выдаваемые $\text{\LaTeX}$ ’ом	94
26.2. Сообщения об ошибках, выдаваемые $\text{\TeX}$ ’ом	96
26.3. Предупреждающие сообщения $\text{\LaTeX}$ ’а	99
26.4. Предупреждающие сообщения $\text{\TeX}$ ’а	99
Термины и определения	101
Алфавитный указатель	104
Список литературы	111