



С. М. ЛЬВОВСКИЙ

Набор и верстка

в пакете **L^AT_EX**

```
\xymatrix{
&& M' \ar@{o->}[dl]^e \ar@/_1pc/@{-->}[ddl]_u \\
& K \ar[rr]^f \ar[dr]^h && L \ar[ul]_a \ar[dl]_g \\
L' \ar@{o->}[ur]_d \ar@/_1pc/@{-->}[rrrr]_v && & & K' \\
& M \ar[rr]^p \ar[ll]_c && & K' \ar@{o->}[ul]_b \\
}
```

С. М. ЛЬВОВСКИЙ

НАБОР И ВЁРСТКА
в системе L^AT_EX

Издание пятое, переработанное

Москва
Издательство МЦНМО
2014

УДК 681.322
ББК 32.97
Л89

Львовский С. М.

Л89 Набор и вёрстка в системе \LaTeX . — 5-е изд., переработанное. — М.: МЦНМО, 2014. — 400 с.

ISBN 978-5-4439-0239-5

Книга посвящена популярной издательской системе \LaTeX , предназначенной для набора и верстки научно-технических текстов с формулами, таблицами, диаграммами любого уровня сложности. В настоящем издании книга существенно переработана: исключен ряд разделов, утративших актуальность, но добавлен материал о \BibTeX 'е, \MetaPost 'е, подготовке презентаций и пр.

Книга будет полезна всем, кто имеет дело с изготовлением на компьютере оригинал-макетов, а также авторам, самостоятельно набирающим научные тексты.

ББК 32.97

ISBN 978-5-4439-0239-5

© Львовский С. М., 2003, 2006, 2014.
© МЦНМО, 2014.

Оглавление

Предисловие	6
I Элементарное введение	9
1. Как проходит работа с системой L ^A T _E X	9
2. Основные понятия	10
3. Набор формул в простейших случаях	21
4. ОбщеT _E X'овские обозначения для выключных формул . .	26
5. Разбиение исходного файла на части	27
6. Обработка ошибок	29
7. Как читать книгу дальше?	36
II Как набирать формулы	38
1. Таблицы спецзнаков с комментариями	38
2. Важные мелочи	51
3. Набор матриц	63
4. Одно над другим	67
5. Тонкая настройка	80
III Набор текста	88
1. Специальные типографские знаки	88
2. Подчеркивания, рамки	91
3. Промежутки между словами	91
4. Диакритические знаки и прочее	94
5. Смена шрифтов в тексте	96
6. Абзацы	102
7. Специальные абзацы	115
8. Сноски	123
9. Между абзацами	125
10. Линейки	134
11. Для любознательных: абзацы нестандартной формы	137

IV	Оформление текста в целом	141
1.	Начнем с главного	141
2.	Классы, пакеты и классовые опции	142
3.	Стиль оформления страницы	146
4.	Поля, размер страницы и прочее	147
5.	Рубрикация документа	151
6.	Эпиграфы	156
7.	Титул, оглавление и пр.	158
8.	Предметный указатель	162
9.	Иллюстрации и таблицы	172
10.	Еще о метках и ссылках	181
11.	Заметки на полях (маргиналии)	184
V	Печать текста с выравниванием	186
1.	Окружение <code>tabbing</code>	186
2.	Таблицы	191
3.	Примеры	198
4.	Дополнительные возможности	205
VI	Создание новых команд	213
1.	Макроопределения	213
2.	Счетчики	226
3.	Окружения типа «теорема»	235
4.	Параметры со значением длины	239
5.	Создание новых окружений: общий случай	242
VII	Блоки и клей	246
1.	Текст состоит из блоков	246
2.	Команды \LaTeX 'а для генерации блоков	247
3.	Команда <code>\hbox</code>	255
4.	Команда <code>\vbox</code>	265
5.	Блочные переменные	266
VIII	Модификация стандартных классов	269
1.	С чего начать	270
2.	Снова о счетчиках	272
3.	Рубрикация	276
4.	Оглавление, список иллюстраций и прочее	284
5.	Перечни общего вида	292
6.	Колонтитулы	299
7.	Плавающие объекты	308

8.	Разное	314
----	------------------	-----

Приложения

А.	Архитектура $\text{T}_\text{E}\text{X}$ 'а и $\text{L}^\text{A}\text{T}_\text{E}\text{X}$ 'а	320
Б.	Библиографии в $\text{L}^\text{A}\text{T}_\text{E}\text{X}$ 'е: $\text{VibT}_\text{E}\text{X}$	327
Г.	Гиперссылки в pdf-файлах	335
Д.	Диаграммы (пакет $\text{X}_\text{Y-pic}$)	337
И.	Интернационализация $\text{L}^\text{A}\text{T}_\text{E}\text{X}$ 'а	343
К.	Классы документов AMS	353
М.	Метапост	355
О.	Откуда взять $\text{T}_\text{E}\text{X}$?	367
П.	Презентации в $\text{L}^\text{A}\text{T}_\text{E}\text{X}$ 'е	369
Р.	Рисунки с помощью подручных средств	371
Ц.	Цвет в $\text{L}^\text{A}\text{T}_\text{E}\text{X}$ 'е	375
Ч.	Что читать дальше	377

Литература	379
-------------------	------------

Предметный указатель	381
-----------------------------	------------

Предисловие к пятому изданию

За 11 лет, прошедшие с выхода последнего переработанного издания этой книги, в компьютерном мире вообще и в \TeX 'овском мире в частности многое изменилось. Возможно, самое главное изменение состоит в том, что значительно возросли возможности поиска в интернете и объем содержащейся там информации. В связи с этим в новом издании я в целом ряде случаев опускаю подробности, ограничиваясь указанием на то, как их может найти заинтересованный читатель.

В отличие от предыдущих изданий, в этом я решил отдать предпочтение штатным средствам перед «самоделками»; в частности, полностью изгнаны рассказы о нестандартных русификациях \LaTeX 'а.

Книга подверглась не только сокращениям: в нее добавлен и новый материал, в частности рассказ о тесно связанных с \LaTeX 'ом системах VibTeX и MetaPost .

В любом из изданий этой книги мой долг — выразить глубокую благодарность трем людям: А. Шеню, И. А. Маховой и В. Д. Арнольду. При подготовке настоящего издания мне были чрезвычайно полезны беседы с А. С. Мелик-Егановым, В. Ю. Радионовым (совместно с которым, в частности, написано приложение А) и В. В. Шуваловым, ценные советы М. Н. Вялого, Г. А. Мерзона и Д. Е. Щербакова (который также любезно согласился написать для настоящего издания приложение П) и нелицеприятная критика Т. Л. Коробковой. Этим своим друзьям и коллегам я с радостью говорю спасибо.

С. Львовский

Предисловие

Замечательный программист и математик Дональд Кнут опередил свое время. Он разработал векторные шрифты (в формате METAFONT) за несколько лет до появления получивших всеобщее признание шрифтов в формате Postscript — но именно постскриптовские шрифты являются сейчас мейнстримом. Он разработал формат файлов (dvi — от $\text{DeVice Independent}$, независимый от устройства), предназначенный для представления сложных текстов в таком виде, чтоб их можно было просмотреть или напечатать под любой операционной системой, на любом компьютере или принтере, задолго до появления формата pdf — однако сейчас для этих целей используется именно pdf . А вот созданная им система компьютерной верстки \TeX (произносится «тех»), для нужд которой Кнут разрабатывал эти шрифты и этот формат файлов, и по сей

день жива, популярна и помирать не собирается, хотя по меркам компьютерного мира она очень стара: ее первая версия появилась еще в 1978 году, в 1989 году она приобрела современные очертания, и с тех пор мало изменилась.

Кое-что в \TeX 'е способно отпугнуть современного пользователя. В самом деле, \TeX не является системой типа WYSIWYG (What You See Is What You Get): чтобы посмотреть, как будет выглядеть на печати набираемый текст, надо запустить отдельную программу. Кроме того, чтобы в \TeX 'е работать, надо потратить определенное время на его изучение: трудно представить себе книгу под названием « \TeX for dummies» (« \TeX для болванов»).

А вот краткий перечень достоинств \TeX 'а.

- 1) Никакая другая из существующих в настоящее время издательских систем не может сравниться с \TeX 'ом в полиграфическом качестве текстов с математическими формулами.
- 2) \TeX реализован на всех современных компьютерных платформах, и все эти реализации действительно работают одинаково.
- 3) Благодаря этому \TeX стал международным языком для обмена математическими и физическими статьями: набрав свою статью в \TeX 'е, математик может послать ее по электронной почте своему коллеге или в редакцию журнала, даже если отправитель работает под Windows, а получатель — под UNIX'ом или, допустим, на Макинтоше.
- 4) Не знаю, уместно ли это называть достоинством, но при подаче статьи в большинство математических и физических журналов требуется набрать ее в \TeX 'е.
- 5) Наконец, основные реализации \TeX 'а для всех платформ распространяются бесплатно.

Разумеется, у \TeX 'а есть и недостатки. Главный из них — в том, что с помощью \TeX 'а тяжело готовить тексты со сложным расположением материала на странице (наподобие рекламных буклетов). Для таких приложений, практически не встречающихся в научно-технической литературе, лучше пользоваться другими программами.

Кроме того, — повторюсь — в \TeX 'е работа с исходным текстом и просмотр того, как текст будет выглядеть на печати, — разные операции. На взгляд автора, благодаря этой особенности время на подготовку текста типографского качества только сокращается, но представления об удобстве у всех разные.

Настоящее пособие посвящено популярной издательской системе, основанной на $\text{T}_\text{E}\text{X}$ 'е, — системе $\text{L}_\text{A}\text{T}_\text{E}\text{X}$. В первую очередь оно предназначено для читателя, которому необходимо по роду своей работы готовить (писать или редактировать) тексты с формулами.

В идеале работа с $\text{L}_\text{A}\text{T}_\text{E}\text{X}$ 'ом должна бы выглядеть так: научный работник пишет на $\text{L}_\text{A}\text{T}_\text{E}\text{X}$ 'е (а редактор редактирует) текст с формулами, после чего файл поступает в опытные руки верстальщика, который доводит его до должного полиграфического качества. На практике автору или редактору порой приходится брать на себя часть «техредовской» работы. Имея это в виду, я рассказываю в книге несколько больше абсолютно необходимого минимума, так что книга может быть полезна и профессиональному полиграфисту, начинающему освоение $\text{L}_\text{A}\text{T}_\text{E}\text{X}$ 'а.

Эта книга выросла из попытки расширить и дополнить книгу [4] — выполненный А. Шенем пересказ с немецкого языка пособия [3] (с его дополнениями). Из исходного текста [3] позаимствованы макросы для печати $\text{T}_\text{E}\text{X}$ 'овских примеров.

Глава I

Элементарное введение

1. Как проходит работа с системой ЛАТЭХ

Как уже отмечалось в предисловии, Т_ЭХ (произносится «тех», пишется также «ТеХ») — это созданная американским математиком и программистом Дональдом Кнутом (Donald E. Knuth) система для верстки текстов с формулами. Сам по себе Т_ЭХ представляет собой специализированный язык программирования (Кнут не только придумал язык, но и написал для него транслятор, причем таким образом, что он работает совершенно одинаково на самых разных компьютерах), на котором пишутся издательские системы, используемые на практике. Точнее говоря, каждая издательская система на базе Т_ЭХ'a представляет собой пакет макроопределений (макропакет) этого языка. В частности, ЛАТЭХ (по-русски произносится «латех», пишется также «LaTeX») — это созданная Лесли Лэмпортом (Leslie Lamport) издательская система на базе Т_ЭХ'a.

Далее в этой книге мы будем, если не оговорено противного, употреблять слова Т_ЭХ и ЛАТЭХ вперемешку. При первом чтении можно воспринимать их как синонимы (на самом деле мы пишем Т_ЭХ, когда речь идет об общих свойствах систем на базе Т_ЭХ'a, а не о специфике именно ЛАТЭХ'a).

Для начала автор должен подготовить с помощью любого текстового редактора файл¹ с текстом, оснащенным командами для ЛАТЭХ'a. Такие файлы по традиции имеют расширение `tex` (описанию того, что должно быть в таком файле, и посвящена вся эта книга). Дальнейшая работа протекает в два этапа. Сначала надо обработать файл с помощью программы-транслятора. В современных версиях Т_ЭХ'a в результате трансляции получается `pdf`-файл. Кроме того, предусмотрена возможность получить файл с расширением `dvi` (device independent —

¹Это должен быть именно «чистый» текстовый файл, а не `doc` или `rtf`.

не зависящий от устройства). В более старых версиях Т_ЕX'а генерация pdf-файлов не предусмотрена, но возможность создать dvi-файл была в Т_ЕX'е всегда.

С полученным pdf-файлом можно делать все то же, что обычно делают с pdf-файлами: просматривать, печатать, вести поиск и т. п. Если же получился dvi-файл, то его необходимо обрабатывать с помощью программ, называемых dvi-драйверами (и входящих в поставку Т_ЕX'а): распечатывать на принтере, смотреть на экране (в таком же виде, как он появится на печати) и т. д. (для разных устройств есть разные драйверы). Неудовлетворенный результатом, автор вносит изменения в исходный файл — и цикл повторяется.

На самом деле повторений цикла будет больше, так как придется еще исправлять синтаксические ошибки в исходном тексте.

Перед тем как начать работать в системе Л^AT_ЕX, вам необходимо уяснить для себя, что нужно сделать, чтобы оттранслировать исходный текст (т. е. создать из него pdf- или dvi-файл); если вы создаете именно dvi-файл, то надо также понять, что нужно сделать, чтобы просмотреть его на экране или напечатать. Кроме того, для создания исходного текста нужно, естественно, уметь обращаться с каким-нибудь текстовым редактором.

Во многих современных текстовых редакторах предусмотрена интеграция с Л^AT_ЕX'ом, позволяющая проводить указанные выше манипуляции с tex- и dvi-файлами одним щелчком мыши или одной клавиатурной командой; в некоторых редакторах можно даже набирать наиболее частые Т_ЕX'овские команды с помощью меню, что, будь то к добру или к худу, приближает интерфейс такой интегрированной среды к интерфейсу редактора типа Word.

2. Основные понятия

2.1. Исходный файл

Исходный файл² для системы Л^AT_ЕX представляет собой собственно текст документа вместе со *специсмволами* и *командами*, с помощью которых системе передаются указания касательно размещения текста. Этот файл можно создать любым текстовым редактором, но при этом необходимо, чтобы в итоге получился так называемый «чистый» текстовый файл. Это означает, что текст не должен содержать шрифтовых выделений, разбивки на страницы и т. п.

Исходный текст документа не должен содержать переносов (Т_ЕX делает их сам). Слова отделяются друг от друга пробелами, при этом

²Для нетерпеливых: пример такого файла приведен на с. 14.

TeX не различает, сколько именно пробелов вы оставили между словами (чтобы вручную управлять пробелами, есть специальные команды, злоупотреблять которыми не советуем). Конец строки также воспринимается как пробел. Соседние абзацы должны быть отделены друг от друга пустыми строками (в произвольном количестве, важно, чтоб была хоть одна).

В правой колонке приведен фрагмент исходного текста, а в левой — то, как он будет выглядеть на печати после обработки системой L^AT_EX.

Слова разделяются пробелами, а абзацы — пустыми строками.

Абзацный отступ в исходном тексте оставлять не надо: он получается автоматически.

Слова разделяются пробелами,
а абзацы ---
пустыми строками.

Абзацный отступ в исходном
тексте оставлять
не
надо: он получается
автоматически.

Как мог заметить читатель, тщательно форматировать исходный текст не обязательно.

2.2. Спецсимволы

Большинство символов в исходном тексте прямо обозначает то, что будет напечатано (если в исходном тексте стоит запятая, то и на печати выйдет запятая). Следующие 10 символов:

{ } \$ & # % _ ^ ~ \

имеют особый статус; если вы употребите их в тексте «просто так», то скорее всего получите сообщение об ошибке (и на печати не увидите того, что хотелось). Печатное изображение знаков, соответствующих первым семи из них, можно получить, если в исходном тексте поставить перед соответствующим символом без пробела знак \ (по-английски он называется «backslash»):

Курс фертинга повысился на 7%,
и теперь за него дают \$200.

Курс фертинга повысился на
7\%, и теперь за него
дают \\$200.

Если символ % употреблен в тексте не в составе комбинации \%, то он является «символом комментария»: все символы, расположенные в строке после него (и сам %), TeX игнорирует. С помощью символа % в исходный текст можно вносить пометки «для себя»:

Это пример.

Это % глупый
% Лучше: поучительный
пример.

Скажем вкратце о смысле остальных спецсимволов. Фигурные скобки ограничивают *группы* в исходном файле (см. с. 15). Знак доллара ограничивает математические формулы (см. разд. 3 и гл. II). При наборе математических же формул используются знаки `_` и `^` («знак подчеркивания» и «крышка»). Знак `~` обозначает «неразрывный пробел» между словами (см. с. 91). Со знака `\` начинаются все Т_ЕX'овские команды (см. разд. 2.3). Знаки `#` и `&` используются в более сложных конструкциях, о которых сейчас говорить преждевременно.

Отметим еще, что символы `<` `>` `|` в тексте употреблять можно в том смысле, что сообщения об ошибке это не вызовет, но при этом может напечататься нечто, совсем на эти символы не похожее. Подлинное место для этих символов, так же как и для символов `=` и `+`, `-` математические формулы, о которых речь пойдет позже. Если вы пишете текст по-русски и оформляете его при этом должным образом (см. с. 29), то не употребляйте «просто так» и двойную кавычку `"` (она используется как вспомогательный символ для организации некоторых специфически русских полиграфических эффектов — см. разд. III.1.1 и III.1.2).

2.3. Команды и их задание в тексте

Задание печатного знака процента с помощью последовательности символов `\%` — пример важнейшего понятия Т_ЕX'а, называемого *командой*. С точки зрения их записи в исходном тексте, команды делятся на два типа. Первый тип — команды, состоящие из знака `\` и одного символа после него, не являющегося буквой. Именно к этому типу относятся команды `\{`, `\}`, `\dots`, `\%`, о которых шла речь на с. 11.

Команды второго типа состоят из `\` и последовательности букв, называемой *именем команды* (имя может состоять и из одной буквы). Например, команды `\TeX` и `\LaTeX` генерируют эмблемы систем Т_ЕX и Л^AT_ЕX соответственно. В имени команды, а также между `\` и именем, не должно быть пробелов; имя команды нельзя разрывать при переносе на другую строку.

Сразу же скажем, что использовать русские буквы в именах нельзя (если не предпринимать специальных ухищрений, говорить о которых мы не будем).

В именах команд прописные и строчные буквы различаются. Например, `\large`, `\Large` и `\LARGE` — это три разные команды (задающие, как мы потом увидим, различные размеры шрифта).

После команды первого типа (из `\` и не-буквы) пробел в исходном

тексте ставится или не ставится в зависимости от того, что вы хотите получить на печати:

`$1` или `$ 1?`

`\$1` или `\$ 1?`

После команды из `\` и букв в исходном тексте *обязательно* должен стоять либо пробел, либо символ, не являющийся буквой (это необходимо, чтобы `TeX` смог определить, где кончается имя команды и начинается дальнейший текст). Вот примеры с командой `\slshape` (она переключает шрифт на наклонный):

`2 turtle doves`

`and a partridge in a pear tree.`

`\slshape2 turtle doves`

`\slshape and a partridge
in a pear tree.`

Если бы мы написали `\slshapeand a partridge...`, то при трансляции `TeX` зафиксировал бы ошибку (типичную для начинающих) и выдал сообщение о том, что команда `\slshapeand` не определена.

С другой стороны, если после команды из `\` и букв в исходном тексте следуют пробелы, то при трансляции они игнорируются. Если необходимо, чтобы `TeX` все-таки «увидел» пробел после команды в исходном тексте (например, чтобы сгенерированное с помощью команды слово не сливалось с последующим текстом), надо этот пробел специально организовать. Один из возможных способов — поставить после команды пару из открывающей и закрывающей фигурных скобок `{}` (так что `TeX` будет знать, что имя команды кончилось), и уже после них сделать пробел, если нужно. Иногда можно также поставить команду `\` (backslash с пробелом после него), генерирующую пробел. Вот пример.

Освоить `LaTeX` проще, чем `TeX`.
Человека, который знает систему
`TeX` и любит ее, можно назвать
`TeX`ником.

Освоить `\LaTeX\` проще,
чем `\TeX`. Человека,
который знает систему
`\TeX{}` и любит ее, можно
назвать `\TeX` ником.

В последней строке этого примера мы не создали пробела после команды `\TeX`, чтобы эмблема `TeX`'а слилась с последующим текстом.

2.4. Структура исходного текста

`LaTeX`-файл должен начинаться с команды

`\documentclass`

задающей общий стиль оформления (как говорят, «класс документа»).

Пример:

```
\documentclass{book}
```

Слово `book` в фигурных скобках указывает, что документ будет оформлен, как книга: все главы будут начинаться с нечетных страниц, текст будет снабжен колонтитулами некоторого определенного вида и т. п. Кроме класса `book`, в стандартный комплект \LaTeX 'а входят классы `article`, `amsart` (для оформления статей), `report` (нечто среднее между `article` и `book`) и некоторые другие. Чтобы задать оформление документа с помощью одного из этих классов, надо в фигурных скобках после команды `\documentclass` указать вместо `book` название требуемого класса. Стандартные классы можно менять, можно создавать и новые классы, но пока что будем исходить из стандартных классов.

После команды `\documentclass` могут следовать команды, относящиеся ко всему документу и устанавливающие различные параметры оформления текста, например, величину абзацного отступа (вообще-то все эти параметры определяются используемым классом, но вам может захотеться их изменить). Далее должна идти команда

```
\begin{document}
```

Только после этой команды может идти собственно текст. Если вы поместите текст или какую-нибудь команду, генерирующую текст (например, `\LaTeX`) до `\begin{document}`, то \LaTeX выдаст сообщение об ошибке. Часть файла, расположенная между командами `\documentclass` и `\begin{document}`, называется преамбулой.

Заканчиваться файл должен командой

```
\end{document}
```

Даже если после `\end{document}` в файле написано что-то еще, \LaTeX это проигнорирует.

Вот составленный по всем правилам \LaTeX -файл.

```
\documentclass{article}
\begin{document}
Hello world.
\end{document}
```

Если вы до сих пор не работали с системой \LaTeX на компьютере, сейчас стоит попробовать. Учтите только, что если вы собираетесь попробовать по-русски, то между `\documentclass` и `\begin{document}` надо обязательно написать кое-что еще — см. с. 29.

2.5. Группы

Важным понятием Т_ЭХ'а является понятие *группы*. Чтобы понять, что это такое, рассмотрим пример.

При обработке Т_ЭХ'ом исходного файла набор текста в каждый момент идет каким-то вполне определенным шрифтом (он называется текущим шрифтом). Команда `\slshape`, с которой мы уже столкнулись в разд. 2.3, переключает текущий шрифт на наклонный, а `\upshape` выполняет обратное переключение; команды `\bfseries` и `\mdseries` меняют жирность шрифта. Подробнее о переключении шрифтов будет рассказано в разделе III.5.

Полужирный шрифт начнется с **этого слова**. Снова светлый, теперь *наклонный*, до нового переключения; *вновь* прямой.

Полужирный шрифт начнется с `\bfseries` этого слова. Снова `\mdseries` светлый, теперь `\slshape` наклонный, до нового переключения; *вновь* `\upshape` прямой.

В этом примере можно обойтись и без команд `\mdseries` и `\upshape` (отменяющих действие предыдущих команд). Для этого часть текста, которую вы хотите оформить полужирным или наклонным шрифтом, можно заключить в фигурные скобки, и дать команду `\bfseries` или `\slshape` *внутри* этих скобок. Тогда сразу же после закрывающей фигурной скобки Т_ЭХ «забудет» про то, что шрифт переключался, и будет продолжать набор тем шрифтом, который был до скобок:

Полужирным шрифтом набрано только **это** слово; после скобок все идет, как прежде.

Полужирным шрифтом набрано только `{\bfseries это}` слово; после скобок все идет, как прежде.

Сами по себе фигурные скобки на шрифт не влияют — они ограничивают *группу* внутри файла. Как правило, задаваемые командами Т_ЭХ'а изменения различных параметров (в нашем случае — текущего шрифта) действуют в пределах той группы, внутри которой была дана соответствующая команда; по окончании группы (после закрывающей фигурной скобки, соответствующей той фигурной скобке, что открывала группу) все эти изменения забываются и восстанавливается тот режим, который был до начала группы. Проиллюстрируем все сказанное следующим примером, в котором используется еще команда `\itshape` (она переключает шрифт на курсивный):

Сначала переключим шрифт на курсив; теперь сделаем шрифт еще и **полужирным**; посмотрите, как восстановится шрифт после конца группы.

Сначала {переключим шрифт на `\itshape` курсив; теперь сделаем шрифт еще и `{\bfseries` полужирным;} посмотрите, как восстановится шрифт после кон{ца г}руппы.

Как видите, группы могут быть вложены друг в дружку. Обратите внимание, что внутри внешней группы курсив начался не с того места, где была открывающая скобка, а только после команды `\itshape` (именно команда, а не скобка, переключает шрифт). Шутки ради мы создали еще одну группу из двух последних буквы слова *конца*, первой буквы слова *группы* и пробела между ними; на печати это никак не отразилось, поскольку внутри скобок мы ничего не делали.

Трюк с постановкой пары скобок `{ }` после имени команды, о котором шла речь на с. 13 — тоже пример использования групп. В этом случае скобки ограничивают «пустую» группу; ставятся они в качестве не-букв, ограничивающих имя команды и при этом никак не влияющих на печатный текст.

Фигурные скобки в исходном тексте (кроме скобок, входящих в состав команд `\{` и `\}`) должны быть сбалансированы: каждой открывающей скобке должна соответствовать закрывающая. Если вы почему-либо не соблюли это условие, при трансляции вы получите сообщение об ошибке.

Некоторые команды, называемые *глобальными*, сохраняют свое действие и за пределами той группы, где они были употреблены. Всякий раз, когда идет речь о глобальной команде, это будет специально оговариваться. Впрочем, глобальных команд в \LaTeX 'е мало, и дойдем мы до них нескоро.

2.6. Команды с аргументами

Команды наподобие `\LaTeX` или, скажем, `\bfseries` действуют «сами по себе»; многим командам, однако, необходимо задать *аргументы*. Первый пример тому дает команда `\documentclass`: слово, указываемое в фигурных скобках, — ее аргумент; если его не указать, то произойдет ошибка. В \LaTeX 'е аргументы команд бывают обязательные и необязательные. Обязательные аргументы задаются *в фигурных скобках*³; если для команды предусмотрено наличие обязательных аргументов, она без

³В некоторых случаях эти фигурные скобки можно опускать — см. с. 24.

них правильно работать не будет. У многих команд предусмотрены также и необязательные аргументы: они влияют на работу команды, коль скоро они указаны, но их отсутствие не ведет к сообщению об ошибке. Необязательные аргументы задаются в *квадратных скобках*.

В частности, у команды `\documentclass` предусмотрен обязательный аргумент, о котором уже шла речь, и необязательный: в квадратных скобках перед обязательным аргументом можно указать список (через запятую) так называемых классовых опций, т. е. дополнительных особенностей оформления. Например, если мы хотим, чтобы книга набиралась шрифтом кегля 12 вместо кегля 10, принятого по умолчанию⁴, и притом в две колонки, мы должны начать файл командой

```
\documentclass[12pt,twocolumn]{book}
```

Наряду с классовыми опциями в ЛАТ_EX'е используются и так называемые *стилевые пакеты*. После команды `\documentclass`, начинающей файл, может следовать команда `\usepackage`, в аргументе которой стоит (через запятую) список подключаемых этой командой стиливых пакетов. (Можно использовать и несколько команд `\usepackage`.) Например, первые две строки файла могут быть такими:

```
\documentclass[12pt,twocolumn]{book}
\usepackage{amsmaths,longtable}
```

Здесь пакет `amsmaths` подключается, чтобы использовать в математических формулах дополнительные шрифты, позволяющие напечатать что-нибудь вроде \mathbb{C} , а пакет `longtable` нужен, чтобы иметь возможность набирать таблицы, простирающиеся на несколько страниц. Когда мы говорим: «чтобы воспользоваться этой возможностью, необходимо подключить такие-то стиливые пакеты», мы молчаливо предполагаем, что поставка ЛАТ_EX'а, которой вы пользуетесь, все эти пакеты содержит (важный пакет `amsmaths` в стандартную поставку входит).

Необязательных аргументов может быть предусмотрено несколько; иногда они должны располагаться до обязательных, иногда после. В любом случае порядок, в котором должны идти аргументы команды, надо строго соблюдать. Между скобками, в которые заключены обязательные аргументы, могут быть пробелы, но не должно быть пустых строк.

2.7. Окружения

Еще одна важная конструкция ЛАТ_EX'а — это окружение (*environment*).

Окружение — это фрагмент файла, начинающийся с текста

⁴Примечание для полиграфистов: Т_EX'овский кегль 10 примерно соответствует нашему девятому кеглю (см. с. 19).

Сергей Михайлович Львовский

НАБОР И ВЁРСТКА В СИСТЕМЕ L^AT_EX

Подписано в печать 10.07.2014 г. Формат 70×100/16. Бумага офсетная. Печать офсетная. Печ. л. 25. Тираж 2000 экз. Заказ №

Издательство Московского центра непрерывного математического образования
119002, Москва, Большой Власьевский пер., 11. Тел. (499) 241-72-85.

Отпечатано в типографии ООО «ТДДС-СТОЛИЦА-8»
тел. 8(495)363-48-86. <http://capitalpress.ru>