

# Система TeX/LaTeX

## (конспект)

Составитель: Геннадий М.

### Содержание:

<a href="#">1. Введение</a>	.....
<a href="#">2. Пошаговая работа</a>	.....
<a href="#">3. Боксы и клей</a>	.....
<a href="#">4. Основные понятия</a>	.....
<a href="#">5. Оформление документа в целом</a>	.....
<a href="#">6. Набор текста</a>	.....
<a href="#">7. Формирование абзацев</a>	.....
<a href="#">8. Формирование страниц</a>	.....
<a href="#">9. Математика</a>	.....
<a href="#">10. Выравнивание текста</a>	.....
<a href="#">11. Иллюстрации</a>	.....
<a href="#">12. Переопределение и создание нового</a>	.....
<a href="#">16. Препринты</a>	.....

## 1. Введение

**TeX** - система для верстки текстов с формулами (название, американский математик и программист Дональд Кнут, придумал от греческих "Технология" и "Искусство", поэтому последняя буква читается как русское X).

### Преимущества:

- наивысшее типографское качество при печати;
- сколь угодно сложные математические формулы;
- работает на всех существующих компьютерных платформах;
- гибкие средства для работы с логической структурой текста;

### Недостатки:

- не является системой типа WYSIWYG;
- результат нужно печатать на лазерном или PostScript принтере;

Поэтому при выборе системы подготовки текста нужно все взвесить. В каких-то случаях можно обойтись и **Microsoft Word**.

Необходимость использования TeXa можно обосновать тем, что он :

- язык международного обмена по математике и физике (большинство научных издательств принимают тексты в печать только в этом формате);
- средство обмена в рамках Internet/Intranet (система хранения и доступа к статьям, отчетам и т.д. в формате HTML);

TeX - специализированный язык программирования довольно низкого уровня, но в нем имеются удобные средства макрорасширений.

### Самые распространенные макропакеты:

- Plain TeX** -- разработан самим Кнутом в качестве примера.
- LaTeX** -- создан Лесли Лэмпортом.
- AMS-TeX** -- разработан Американским Математическим Обществом.
- AMS-LaTeX** -- развитие предыдущей версии.

## 2. Пошаговая работа

Как собственно происходит работа с системой LaTeX:

- >name.tex подготовка исходного файла (текст + команды TeX).
- name.tex->name.dvi обработка программой транслятором.
- name.dvi->display просмотр картинки dvi-драйвером.
- name.dvi->name.ps получение универсального формата PostScript.

На каждом шаге происходит выявление ошибок и их исправление.

С конкретной работой лучше познакомиться на практике.

Возникает вопрос, почему нельзя объединить и автоматизировать все шаги и сделать систему типа WYSIWYG?

Во-первых: такие попытки есть (LyX, Scientific Notebook).

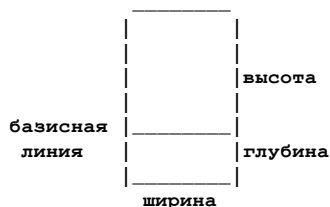
Во-вторых: все равно это будет "игрушечный" вариант.

И объяснением этому являются сложные и оригинальные принципы работы самого транслятора TeX (обычно это излагается в конце, но мне кажется, что лучше сказать об этом сразу):

## 3. Боксы и клей

TeX собирает сложные страницы, начиная с простых символов, собирая их в более крупные и т.д. - это большая работа по склеиванию.

**Бокс - двумерный объект** прямоугольной формы, характеризуется тремя величинами (высота, ширина, глубина):



Все что напечатано TeX-ом на странице склеено из этих простых типов в различных комбинациях.

Два способа склеивания : горизонтальный, вертикальный.

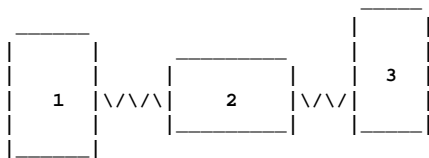
Для наглядности работы TeX можно вспомнить как собирался металлический набор вручную (с клиньями и запорами в старых типографиях). Так же работает TeX, только в виртуальном компьютерном режиме и пробелы между буквами и строками определяются не "на глазок", а по определенному алгоритму с использованием другого ключевого понятия:

**Клей - некоторый магический раствор**, используемый для скрепления боксов друг с другом (хотя сам Кнут отмечал, что название не очень удачно).

Клей имеет три атрибута:

- `space` - естественная величина;
- `stretch` - способность растягиваться;
- `strink` - способность сжиматься;

На самом деле лучше бы пользоваться понятием пружины с коэффициентами жесткости отдельно на растяжение и на сжатие.



Если потянуть за крайние блоки, то конструкция растянется в соответствии с жесткостью пружин (или соответственно сожмется).

Этот процесс называется установкой клея.

Как только клей установлен, бокс становится жестким и неделимым.

Тоже самое происходит и при формировании страницы, т.е. в вертикальном режиме.

В процессе обработки исходного текста TeX, в каждый момент, времени находится в одном из трех **режимов или мод** (на самом деле есть еще "подрежимы"):

- горизонтальная мода (формирование абзацев);
- вертикальная мода (в начале работы или между абзацами);
- математическая мода (обработка математических формул);

Теперь, разобрав основные принципы работы TeX-а, можно понять, что вряд ли можно сделать систему WYSIWYG, адекватную TeX.

---

## 4. Основные понятия

### 4.1 Исходный файл = текст документа + спецсимволы + команды.

Пример самого минимального LaTeX-файла, составленного по всем правилам:

```
\documentstyle{article}
\begin{document}
  Hello, world!
\end{document}
```

Основные правила при наборе текста:

- исходный текст не должен содержать переносов (TeX сделает их сам);
- 'return' -- это то же самое, что пробел;
- два пробела рядом считаются за один пробел;
- абзацы разделяются одной или несколькими пустыми строками;

Интересно, что эти правила противоречивы:

2'сг' = пустая строка = два пробела = один пробел,  
получилось, что один пробел -- признак разделения абзацев.

### 4.2 Специальные символы:

- % комментарии
- { начало группы
- } конец группы
- \$ ввод математики
- \_ нижние индексы математики
- ^ верхние индексы математики
- ~ неразрывный пробел
- \ сигнальный символ (команд)
- # параметры в определениях команд
- & табулятор

### 4.3 Команды.

- а) командное слово - \имякоманды (из букв [A..Z a..z])
- б) командный символ - \\* символ не буква

В первом случае команду надо обязательно как-то отделить от последующего текста (пробелом или другой командой).

Команды могут иметь аргументы: {обязательные}, [необязательные].

Пример: `\documentstyle[12pt,twocolumn]{book}`

Команды могут иметь варианты (со звездочкой) :

Пример: `\section` - начать раздел  
`\section*` - начать раздел не нумеруя

В TeX встроено 300 примитивов и 600 обычных команд.

В LaTeX используются команды еще более высокого уровня, ими пользоваться удобнее и безопаснее, хотя некоторым навязанным образом.

### 4.4 Параметры.

TeX в каждый момент обработки исходного текста учитывает значения различных параметров: величину абзацного отступа, ширину и высоту страницы и многое другое.

Можно сказать, что параметры -- частный случай команд, т.е. это команды которые выдают число.

Пример: `\parindent=2cm` -- задать абзацный отступ.

Кстати основные **единицы измерения:**

sp (scaled point)	-	масштабный пункт (основа всех целочисл. расчетов)
pt (point)	-	пункт ( 1pt = 65536sp = 0.35mm)
pc (pica)	-	пика ( 1pc = 12pt )
in (inch)	-	дюйм ( 1in = 72.27pt )
cm (centimeter)	-	сантиметр ( 2.54cm = 1in )
mm (millimetr)	-	миллиметр ( 10mm = 1cm )

Можно записывать дробные размеры с помощью точки или запятой.

Обязательно указание 0pt или 0mm.

Удобно также пользоваться относительными единицами:

em - примерно ширина буквы М  
ex - примерно высота буквы x

### 4.5 Группы.

Это еще одно важнейшее понятие TeX-а, как впрочем и любого другого языка программирования, где оно известно как "блочная структура". Внутри задаются локальные переменные и определения которые не видны извне.

Пример:

- а) `{\bf Это}` слово полужирно.
- б) `\centerline{Эта информация должна быть {\it в центре}}`  
В последнем примере два набора скобок имеют существенно различные функции.

### 4.6 Окружения(environment).

Эта конструкция объединяет в себе принципы предыдущих:

```
\begin{Имя окружения} - начальные действия
|-----
|
| Тело
|
|-----
\end{Имя окружения} - конечные действия
```

Пример: `\begin{center}`  
Все строки этого абзаца  
будут центрированы  
и переносов слов не будет  
`\end{center}`

В дальнейшем мы узнаем основные стандартные команды и т.д., а также научимся их переопределять и создавать новые.

---

## **5. Оформление документа в целом**

Пользоваться непосредственно самим TeX-ом сложно, но как раз поэтому существует LaTeX, который освобождает вас от многих забот об оформлении документа. Хотя при этом навязывает некоторый порядок, который может и не нравиться.

Конечно, любой стандарт можно модифицировать и даже создать свой собственный.

LaTeX предоставляет 4 основных стиля:

**letter** - для оформления деловых писем как это принято в США.

- article** - для оформления статей.
- report** - для более крупных статей, разбитых на главы.
- book** - документ будет оформлен как книга: все главы начинаются с нечетной страницы, текст снабжен колонтитулами определенного вида.

Приведем табличку в которой описаны основные отличия (по умолчанию) первых трех стилей :

	article	report	book
Разбиение на главы	-	+	+
Двухсторонняя печать*	-	-	+
Титульный лист	-	+	+
Колонтитулы	-	+	+
Одинаковая высота всех страниц	-	-	+

Здесь: '+' -- возможность присутствует  
 '-' -- возможность отсутствует  
 '\*' -- с разными полями для четных и нечетных страниц

### Напишем простенький документ:

```
\documentstyle[12pt,russian]{article}
%
% Такая команда использовалась в старой версии (LaTeX 2.09),
% новая версия совместима с предыдущей. Основные новшества:
% - расширение набора шрифтов
% - включение в себя формата AmSTeX ..
% - настройка опций на форматы листов
% - добавлены новые команды
% Чтобы получить доступ к новым возможностям надо заменить введение на
% \documentclass[a4paper,12pt]{article}
% \usepackage{russian} - пакет поддержки русского языка
% \Russian - переключение на переносы рус. слов и т.д. ( обратно: \English)

\begin{document}

% Работая с LaTeX-ом разумно делать заголовки и нумерацию
% разделов не вручную , а с помощью специальных команд.
\author{...}
\title{...}
\maketitle{...}
\tableofcontents % создает файл *.toc с информацией о содержании

\section{...} % или если необходимо более мелкое деление
% subsection-subsubsection-paragraph-subparagraph
% Идет соответствующая нумерация, хотя все зависит от стиля и от того,
% есть ли звездочка при команде.

\begin{...} % Кроме команд разбивающих документ на части,
\item ... % очень полезными для улучшения зрительного восприятия
\item ... % текста является создание перечней:
\begin{...} % В LaTeX-е описаны три вида перечней (окружения):
\item ... % itemize - простые ( "горошина" - * . )
\item ... % enumerate - нумерованные ( 1 а I(римские) A)
\end{...} % description - описательные (конкретные названия)
\end{...}

\label{mylabel} % В LaTeX-е можно сослаться на практически любой
. % элемент текста : на страницу, на раздел документа
. % на номер рисунка, на номер элемента в нумерованном
. % перечне.
Как указано на стр~\pageref{mylabel} ...

% По поводу размещения иллюстраций, псевдографики и графики
% будет отдельный раздел.

\begin{thebibliography}{99} % А в тексте можно сослаться на этот
\bibitem{book1} % список литературы (занумерованный автом.)
\bibitem{book2} % с помощью команды \cite{...}
\bibitem{book3} %
\end{thebibliography} %

\end{document}
```

## 6. Набор текста

### 6.1 Черточки.

Большинство знаков препинания (точка, запятая, двоеточие) набирается очевидным образом, но некоторые требуют специального набора:

- дефис или знак переноса  
-- черточка или en-тире  
--- тире или em-тире  
\$-\$ знак минус

## 6.2 Кавычки.

В книгах существует их два вида: открывающие и закрывающие. Похожие символы есть на стандартных клавиатурах, они и используются при наборе. Для получения знака двойной кавычки (т.е. лигатуры) надо просто ввести их дважды:  
`` открывающие  
'' закрывающие.

## 6.3 Многоточия.

Набранные подряд три точки (...) для пропорциональных шрифтов окажутся расположены слишком близко. Поэтому используются специальные команды:  
a) \ldots нижнее многоточие  
b) \cdots центрированное (обычно в математике между +, -, и т.д.)  
c) \vdots вертикальное (обычно в математике)  
d) \ddots диагональное (в матрицах и т.д.)

## 6.4 Прочие значки (кроме математики) и т.д.

\S - знак номера параграфа  
\dag - кинжал или обелиск  
\copyright - знак авторского права  
\pounds - знак фунта стерлингов

Конечно в тексте можно использовать любой мат. символ (заклучив его в \$...\$), например Я \$\heartsuit\$ тебя.

Это слово будет \underline{подчеркнуто}.  
Два слова будут \fbox{в рамке}.

Диакритические знаки - дополнительные значки, размещающиеся над или под буквой (для разных европейских языков).

\' - постановка ударения  
\" - для русской буквы 'е'

Пример:  
\"Ежик под \"елкой.

Здесь небольшая тонкость: если у LaTeX-овской команды один обязательный аргумент и после команды непосредственно следует буква, то она и воспринимается как аргумент (поэтому {} не надо).

## 6.5 Промежутки между словами:

\enskip - пробел в 0.5em  
\quad - пробел в 1em  
\qquad - пробел в 2em  
\hspace{длина} - конкретный промежуток  
\, - тонкий пробел (шпация)  
\: - средний пробел  
\; - толстый пробел

## 6.6 Шрифты:

Перечислим все существующие в стандартном комплекте LaTeX-ха гарнитуры (виды шрифта) и кегли (размеры). Но не увлекайтесь переключением шрифтов! Чем их меньше, тем легче читать текст.

### Гарнитуры:

\bf (boldface) - полужирный  
\it (italic) - курсив  
\sl (slanted) - наклонный  
\sf (sans serif) - рубленый  
\sc (Small Caps) - Капитель  
\tt (typewriter) - имитация пишущей машинки  
\rm (roman) - прямой светлый

### Кегли:

\tiny - малюсенький  
\scriptsize - очень маленький (как индексы)  
\footnotesize - маленький (как сноски)  
\small - мелкий  
\normalsize - нормальный  
\large - большой  
\Large - очень большой  
\LARGE - совсем большой  
\huge - громадный  
\Huge - грандиозный

Реальный размер шрифтов зависит от выбранного стиля (10pt..12pt..).

В новейших версиях LaTeX-а используется NFSS (New Font Selection Scheme) новая схема выбора шрифтов. Их стало гораздо больше за счет большей гибкости определения, а именно, каждый шрифт в NFSS характеризуется

четырьмя параметрами:  
 семейство (family) - roman ,рубленный и т.д.  
 гарнитура (series) - "жирность" и плотность  
 начертание (shape) - прямой, курсивный и т.д.  
 размер (size) -

По прежнему поддерживаются привычные команды, но смысл их несколько иной.  
 Более серьезные различия - использование "некурсивных" шрифтов в  
 мат. формулах: вместо  $\{ \rm xyz \}$  надо  $\{ \mathrm{xyz} \}$ .

## 7. Формирование абзацев

Одна из главных обязанностей систем набора текстов -- это взять длинную последовательность слов и разбить ее на строки подходящего размера (горизонтальная мода).  
 TeX будет искать самый лучший способ напечатать каждый абзац в соответствии с принципом наименьшей "плохости".

Плохость ("badness") - некоторое числовое значение, которое TeX присваивает каждой строке, что бы оценить эстетическое восприятие пробелов между словами.

Есть некоторые правила у TeX по переносам:

- не делает автоматических переносов в словах с диакритическими знаками, а также в которых присутствуют цифры, знаки препинания и т.д.;
- если в слове есть дефис то на нем и произойдет перенос (например "генерал-губернатор");

Но компьютер не понимает мотивы поведения людей и иногда ему необходима конкретная подсказка (намек):

**Указать перенос в слове:**

- a) \- "одноразовый (на это слово)" способ : на\-"ем\"-ник
- b) \hyphenation "глобальный (на весь документ)" способ.  
Пример: \hyphenation{вклю-чен об-ласть}

**Запретить перенос в слове:**

- a) \mbox (make box) -"одноразовый" способ.  
Пример: Параметр \mbox{filename} задает имя файла.
- b) \- "одноразовый, хулиганский" способ, поставив команду в конце слова.  
Пример: Это слово корова\ - перенесено не будет.
- c) \hyphenation{...корова,...} "глобальный" способ: указать слово без дефисов.

**Указать разрыв между словами:**

- a) \\ принудительный разрыв строки без ее растяжения.  
Пример: Эта строка\\ была разорвана.
- b) \linebreak принудительный разрыв строки и ее дальнейшее выравнивание по правому краю.  
Пример: Эта строка была \linebreak разорвана

**Запретить разрыв между словами:**

- ~ символ неразрывного пробела или "связка" -- ключ к успешному разбиению строк. Необходимо использовать в следующих случаях (все эти правила требуют семантического анализа и не могут быть автоматизированы):
- a) После сокращений, чтобы точка не воспринималась концом предложения:  
См.~Рис.~5
- b) В ссылках на именованные части документа:  
Глава~12                   Рисунок~3  
Приложение~А            Теорема~1.2
- c) Между именами собственными и между частями сложной фамилии:  
Дональд~Е. Кнут       Петр~I
- d) Между математическими символами, присоединенными к существительным:  
строка~\$s\$   длиной~\$l\$
- e) Между символами в ряде:  
1,~2 или~3  
1,~2, \ldots,~\$n\$
- f) Когда символ тесно связан с предлогом:  
от~\$x\$                   увеличить \$z\$ на~1  
от 0 до~1
- g) Когда математическое предложение выражено словами:  
равно~\$n\$                   для всех больших~\$n\$  
меньше чем~\$\epsilon\$
- h) Когда случаи поименованы внутри абзаца:  
(b)~Покажите, что \$f(x)\$ является  
(1)~непрерывной; (2)~ограниченной

## 8. Формирование страниц

До сих пор мы рассматривали, что происходит с документом "на уровне строки" (горизонтальная мода). Теперь изменим масштаб рассмотрения: как из абзацев формируется страница(вертикальная мода).

### 8.1 Закончить абзац (эквивалентные варианты):

- пустая строка
  - команда \par
- Оба способа, выполненные подряд, дополнительных промежутков не создадут

(т.е. команда \rag в вертикальной моде ничего не делает).

## 8.2 Задать "свой" промежуток между абзацами:

```
\smallskip    маленький вертикальный пробел.
\medskip     вертикальный пробел побольше.
\bigskip     еще больше (точные размеры зависят от стиля и кегля).
\vspace{...} промежуток конкретного размера.
Можно задать промежуток не фиксированной, а переменной длины:
\vspace{x plus y minus z}, здесь:
    x,y,z     - длины
    plus,minus - ключевые слова (без backslash)
```

Говоря Техническим языком \vspace вставляет в страницу клей.

## 8.3 Интерлиньяж -- интервал между строками (устанавливается

автоматически для каждого шрифта). Иногда бывает необходимо пропорционально изменить все интервалы:

```
\renewcommand{\baselinestretch}{1.01} -- увеличит на 1%
```

## 8.4 Разбиение на страницы в TeX -- процесс "одноразовый" (абзац

TeX читает целиком, а затем перебирает различные способы разбиения на строки и выбирает из них оптимальный). Такой метод для страниц слишком "ресурсоемок".

Поэтому как только TeX набирает достаточно строк он производит разрыв. Выбор обычно невелик (за счет клея между абзацами), поэтому надо пользоваться специальными командами:

### Принудительный разрыв страницы:

```
\newpage     -- дополняется снизу пустым пространством;
\clearpage   -- и еще допечатывает плавающие иллюстрации;
\pagebreak   -- с "растягиванием" страницы;
Кстати, чтобы создать пустую страницу надо TeX обмануть:
\newpage \mbox{} \newpage
```

### Запрет разрыва страницы:

```
\nopagebreak -- локальный т.е. между конкретными абзацами;
\samepage    -- глобальный т.е. разрывы станут возможны только между абзацами, а не внутри и не между текстом и выключной формулой;
```

## 8.5 Набор в две колонки:

- [twocolumn] -- стилевая опция для всего документа;
- \twocolumn[широкий текст] -- сначала выполнить команду \clearpage, затем (если надо) широкий текст, затем начать формировать две колонки (\onescolumn - обратное переключение);
- Как сделать две колонки в конкретном месте?
  - отменить действие команды \clearpage
  - использовать окружение minipage
  - использовать команду \parbox
  - ..... ?

---

# 9. Математика

## 9.1 Основные способы задания формул:

- "внутренних" т.е. внутри текста:
  - \$...\$ -- стандартный способ TeX;
- "выключных" т.е. выделенных в отдельную строку:
  - \$\$...\$\$ -- стандартный способ TeX;
  - {equation} -- способ LaTeX с автоматической нумерацией (стилевые опции: [leqno] - нумерация слева, [flegn] - формулы слева).

## 9.2 Основные принципы набора формул:

- пробелы игнорируются (TeX их сделает сам);
- пустые строки не разрешаются;
- мат. формула является группой;
- каждая буква рассматривается как имя переменной и набирается шрифтом "математический курсив";
- поэтому обычный текст включается командой \mbox;

## 9.3 Греческие буквы:

задаются командами по их английским названиям (начинаются с большой буквы для прописных букв \psi \Psi).

\alpha	\iota	\sigma
\beta	\kappa	\varsigma
\gamma	\lambda	\tau
\delta	\mu	\upsilon
\epsilon	\nu	\phi
\varepsilon	\xi	\varphi
\zeta	\pi	\chi
\eta	\varpi	\psi
\theta	\rho	\omega
\vartheta	\varrho	

Прописные греческие буквы не совпадающие по начертанию с латинскими (печатаются прямым шрифтом):

\Gamma	\Delta	\Theta
\Lambda	\Xi	\Pi
\Sigma	\Upsilon	\Phi
\Psi	\Omega	

Если нужны наклонные прописные греческие буквы, то надо, например: {\mit\Sigma};

#### 9.4 Бинарные операции:

+	плюс
-	минус
*	умножение
\times	умножение "крестиком"
\div	деление (минус между точками)

#### 9.5 Бинарные отношения:

<	меньше
>	больше
=	равно
\le	меньше либо равно
\ge	больше либо равно
\ne	не равно
\sim	подобно (одна волна)
\approx	приблизительно (две волны)
\equiv	эквивалентно ("тройное равенство")

#### 9.6 Стрелки различных видов:

\to	тонкая	стрелочка	вправо
\Rrightarrow	двойная	стрелочка	вправо
\gets	тонкая	стрелочка	влево
\Lleftarrow	двойная	стрелочка	влево

#### 9.7 Функции "типа синус":

\sin	\tan	\exp
\cos	\arctan	\dim
\arcsin	\log	\lg
\arccos	\ln	

#### 9.8 "Элементарные" операции:

\sum	сумма	\prod	произведение
\lim	предел	\inf	инфимум
\max	максимум	\int	интеграл
\min	минимум	\oint	контурный интеграл

#### 9.9 Скобки различных видов:

()	круглые скобки
[]	квадратные скобки
\{ \}	фигурные скобки
	знак модуля
\langle \rangle	угловые скобки

Задание размера автоматическое:

\left( ... \right) -- по высоте фрагмента формулы.

Эти команды могут появляться только парами, однако скобку можно сделать невидимой, задав вместо нее точку: \left.

Ограничители не обязаны быть однотипными.

Задание размера явное:

\bigl ... \bigr -- конкретные размеры зависят от

\Bigl ... \Bigr кегля и подбираются экспериментально;

\biggl ... \biggr

\Biggl ... \Biggr

Эти команды не обязаны появляться парами.

#### 9.10 Разные значки:

\partial	-- частная производная
\prime или '	-- штрих-производная
\forall	-- "для всех"
\exists	-- "существование"
\Box	-- квадратик
\Diamond	-- ромбик
\sharp	-- музыкальный диэз
\flat	-- музыкальный бемоль

#### 9.11 Переносы:

а) в выключных формулах:

никогда не переносятся автоматически, но можно это сделать используя некоторые трюки типа окружения {\array}:

```
$$ \begin{array}{l} e^x=1+x+\frac{x^2}{2!}\\\quad +\frac{x^3}{3!}+\cdots \end{array} $$
```

б) во внутритекстовых формулах:

после знаков бинарных отношений и операций, но не может разорваться часть ограниченная фигурными скобками.

#### 9.12 Буквы в формулах набираются:

а) математическим курсивом \mit (по умолчанию) -- эта команда непосредственно используется редко.

б) полужирным прямым шрифтов (если в преамбуле задать \boldmath )

Если нужен другой шрифт надо дать соответствующую команду.

Пример: Обозначение  $\{\mathbf{P}\}^n$  используется для проективного пространства.

#### 9.13 Текст в формулах набирается:

\mbox{текст} , а иначе TeX расставит свои маленькие "математические" пробелы.



Неправильно (вся правая часть сольется в одно слово) :

$$2x = x + x \quad \{\text{для всех } x\}$$

Правильно:

$$2x = x + x \quad \text{для всех } x.$$

Используется шрифт который был текущим перед началом формулы.

#### 9.14 Надстрочные знаки:

это дополнительные значки над буквой

или фрагментом формулы:

a)  $\overline{\dots}$  -- горизонтальная черта над любым фрагментом формулы;

b)  $\overrightarrow{\dots}$  -- стрелка (вектор) над любым фрагментом формулы;

c) "узкие" значки:

$\hat{\phantom{x}}$  - шляпка  
 $\tilde{\phantom{x}}$  - волна  
 $\bar{\phantom{x}}$  - черточка  
 $\vec{\phantom{x}}$  - вектор  
 $\dot{\phantom{x}}$  - точка  
 $\ddot{\phantom{x}}$  - две точки

Пример:  $\vec{a}$

d) "широкие" значки (но не безгранично):

$\widehat{\dots}$   
 $\widetilde{\dots}$

Пример: Тождество  $\widehat{f \cdot g} = \hat{f} \cdot \hat{g}$  & означает...

#### 9.15 Элементарные мелочи:

a) степени и индексы: набираются знаками  $\wedge$  и  $\_$  соответственно:

$R^{i_{jkl}}$  или  $R_{j\{i\}}_{\{kl\}}$

Во втором случае, оформив индексы к пустой формуле, добились чтобы они располагались не один под другим.

b) дроби, обозначаемые косой чертой, набираются непосредственно:

$x+1/x \ge 2$  верно для всех  $x > 0$

c) запятая в десятичной дроби записывается в фигурных скобках, иначе после нее будет поставлен дополнительный пробел:

$\pi \approx 3{,}14$

d)  $\sqrt[\text{показатель}]{\text{подкоренное выражение}}$  -- корень

$\sqrt[3]{x^3} = x$

e)  $\log_{\{\text{основание}\}}\{\text{аргумент}\}$  -- основание задается как нижний индекс.

e) штрихи обозначаются знаком  $'$  и не оформляются как верхние индексы:

$f(g)' = f'g + 2f'g' + fg''$

f) "пределы" у знака суммы (по умолчанию):

- в выключной формуле печатаются "над и под" знаком;

- во внутритекстовой "сбоку", как и индексы;

$\sum_{i=1}^n n^2$

"пределы" у знака интеграла (по умолчанию):

- печатаются как индексы "сбоку";

$\int_0^1 x^2 dx$  -- от 0 до 1 от  $x^2$

$\lim_{x \rightarrow 0} x^2$  -- обязать расположение "над и под";

$\lim_{x \rightarrow 0} x^2$  -- обязать расположение "сбоку";

$\int \lim_{x \rightarrow 0} x^2 dx$

#### 9.16 Одно над другим:

a)  $\frac{\text{числитель}}{\text{знаменатель}}$  -- запись для дроби обыкновенной (одну букву или цифру можно не брать в скобки):

$12 + \frac{x}{2} = \frac{1+x}{2}$

b) горизонтальная фигурная скобка:

$\overbrace{\text{фрагмент формулы}}^{\text{надпись}}$  -- над формулой

$\underbrace{\text{фрагмент формулы}}_{\text{подпись}}$  -- под формулой

c) расположение типа "над-под":

$\overset{\text{верхняя часть формулы}}{\underset{\text{нижняя часть формулы}}{\text{фрагмент формулы}}}$

$\left\{ \begin{matrix} i \\ j \end{matrix} \right\} \left\{ \begin{matrix} k \\ l \end{matrix} \right\}$  здесь {} выполняют две функции.

$\binom{n}{k}$  -- биномиальные коэффициенты

$\sum_{k=0}^n \binom{n}{k}$

d) расположение типа "вровень-над":

$\stackrel{\text{будет над строкой}}{\text{будет в строке}}$

$A \stackrel{\text{будет над строкой}}{\text{будет в строке}} \rightarrow B$ , почти:  $A - f -> B$

#### 9.17 Матрицы

$\begin{array}{\text{преамбула}}$

|-----

| преамбула это ряд букв (по букве на столбец),

| описывающих столбцы:

| c -- центрированы;

| l -- выровнены по левому краю;

| r -- выровнены по правому краю;

| сама матрица формируется с использованием:

| \ \ -- разделяет строки матрицы;

| & -- разделяет элементы столбцов внутри строки;

|-----

$\end{array}$

Пример записи простой квадратной матрицы из n элементов

(скобки надо записать отдельно):

$$\left( \begin{array}{cccc} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{array} \right)$$

$$\left( \begin{array}{cccc} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{array} \right)$$

$$\left( \begin{array}{cccc} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{array} \right)$$

$$\left( \begin{array}{cccc} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{array} \right)$$

$$\left( \begin{array}{cccc} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{array} \right)$$

$$\left( \begin{array}{cccc} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{array} \right)$$

$$\left( \begin{array}{cccc} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{array} \right)$$

$$\left( \begin{array}{cccc} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{array} \right)$$

## 9.18 Системы уравнений: можно записывать используя {array}:

```
$$
\left\{
\begin{array}{rcl}
x^2+y^2 & = & 7 \\
x+y & = & 3
\end{array}
\right.
$$
```

## 10. "Выравнивание" текста

"Пропорциональные" шрифты не позволяют выравнивать, например, колонки текста вводом пробелов. Используют следующие средства:

### 10.1 Имитация табулятора:

```
\begin{tabbing}
|-----
| \=          отметить позицию табулятора;
| \>         перескочить к очередной позиции табулятора;
| \\         закончить строку;
| \kill      закончить строку не печатая, но запомнив табуляторы;
| \pushtabs  запомнить расположение табуляторов;
| \proptabs  восстановить запомненное ранее;
| \'        разместить текст до позиции табуляции;
| \`        прижать конец строки к правому краю;
| \+        увеличить номер "стартового" табулятора;
| \-        уменьшить номер "стартового" табулятора;
| \<        делать как \- для одной этой строки;
| Внутри окружения {tabbing} вместо \= надо \a=e
|                                     вместо \'e надо \'a'e ;
|-----
\end{tabbing}
```

Простой пример:

```
\begin{tabbing}
| начинаем \=продолжаем \=заканчиваем \kill
| \>раз \>два \>три
| \>четыре \>пять \>шесть
\end{tabbing}
```

Другие возможности используются редко, для этого есть более удобное средство, где:

- ширина колонок выбирается автоматически;
- можно разлиновывать таблицы;
- можно делать таблицы сложной структуры:

### 10.2 Верстка таблиц:

```
\begin{tabular}{преамбула}
|-----
| преамбула это нечто, описывающее колонки:
|   c -- центрированы;
|   l -- выровнены по левому краю;
|   r -- выровнены по правому краю;
|   | -- задает вертикальную линию;
|   p{ширина колонки} -- для абзацев текста;
| сама таблица формируется с использованием:
|   \\ -- разделяет строки таблицы;
|   & -- разделяет колонки внутри строки;
|   \hline -- задает горизонтальную линию на всю ширину;
|   \cline{первая-последняя} -- задание линии по колонкам;
|   \multicolumn{N}{P}{T} -- задает "широкую" графу, где:
|     N -- количество "охватываемых" колонок;
|     P -- "преамбула" для этой графы;
|     T -- текст, записываемый в графу;
|-----
\end{tabular}
```

Пример:

```
\begin{tabular}{|l|lr|}
\hline
\multicolumn{2}{|c|}{Западные сладости}\hline
Название & Количество & Цена \\
\hline
Сникерс & штука & 2000 \\
\cline{2-3}
& десяток & 19000 \\
\hline
Марс & штука & 1800 \\
\hline
\end{tabular}
```

## 11. Иллюстрации

**11.1 Сначала надо** задать пустое место не пропадающее при разрыве страницы:

```
\begin{table}[tbph]    -- плавающая таблица
\begin{figure}[tbph]  -- плавающая картинка
|-----
| "tbph" -- пожелание по поводу размещения.
| Это одна (или несколько букв если согласен на любую):
|   t -- разместить в верхней части страницы;
|   b -- в нижней части страницы;
|   p -- на отдельной "иллюстрационной" странице;
|   h -- разместить именно в этом месте;
| По умолчанию задано [tbp]. Все свойства для "таблиц" и "картинок"
| совпадают за исключением подписи.
|-----
\end{figure} или \end{figure}
```

Пример:

```
\begin{figure}
  \vspace{4cm}
  \caption{Белый квадрат}
  \label{my_label}
\end{figure}
```

**11.2 Псевдорисунки** -- "примитивные" (т.е. из простых элементов) рисунки;

- все размеры задаются десятичными числами (возможно дробными через точку);
- не должно быть пустых строк;
- не должно быть пробелов (при переносе использовать знак % );
- параметры регулирующие вид:
  - a) единица измерения `\unitlength=1mm` (по умолчанию =1pt);
  - b) `\thinlines` -- режим тонких линий (по умолчанию);
  - c) `\thicklines` -- режим толстых линий;
  - d) `\linethickness{2.5mm}` -- для ненаклонных линий;

```
\begin{picture}(ширина,высота)
|-----
| \put(X,Y){ОБЪЕКТ}    рисовать графический объект, где:
|   X -- абсцисса (относительно левого нижнего угла);
|   Y -- ордината (также);
|   ОБЪЕКТ:
|     a) ОБЫЧНЫЙ ТЕКСТ
|     b) \line(ВПРАВО,ВНИЗ)(ДЛИНА) -- линия ,где
|        ДЛИНА - длина горизонтальной проекции;
|        - просто длина для вертикального отрезка;
|        ВПРАВО,ВНИЗ - целые числа не имеющие общих
|        делителей и |ВПРАВО+ВНИЗ|<=6 ;
|     c) \vector(ВПРАВО,ВНИЗ)(ДЛИНА) -- стрелочка ,где все
|        как выше только |ВПРАВО+ВНИЗ|<=4 ;
|     d) \circle{ДИАМЕТР} -- окружность или круг (если *);
|        Варианты диаметров ограничены шрифтами и если
|        не найден ,то берется ближайший.
|     e) \oval(ДЛИНА,ШИРИНА)[ЧАСТЬ] -- закругленный прямоугольник.
|        ЧАСТЬ - одна (для половинки) или две
|        (для четвертушки) буквы:
|           t - верхняя;   r - правая;
|           b - нижняя;   l - левая;
| \multiput(X,Y)(DX,DY){N}{ОБЪЕКТ} рисовать несколько регулярно
|        расположенных объектов;
|-----
\end{picture}
```

### 11.3 Графика:

Рекомендуемый способ: использование "почти" стандартного пакета .

```
\usepackage{graphics} -- подключается в преамбуле документа.
                        Умеет вставлять рисунки в форматах PSX ,BMP ,EPS.
```

Рисунки в EPS формате можно масштабировать и вращать (без потери качества) и печатать в многоцветном варианте.

```
\includegraphics[лн_x,лн_y][пв_x,пв_y]{имя_файла}
a) Если указан один необяз. аргумент, то полагается лн_x=лн_y=0;
b) Если отсутствуют оба, то размер определяется автоматически;
c) Вариант команды со звездочкой делает "обрезание".
```

В следующих командах вместо "текста" можно вставить график как единую букву (`\includegraphics`):

```
\rotatebox{угол_в_градусах}{текст} - помещает текст в бокс подобно
команде \mbox{} и при этом его разворачивает.
\resizebox{ширина}{высота}{текст} - масштабирование; вместо одного из
параметров <ширина> или <высота> можно использовать "!" ,чтобы
сохранить аспектное отношение.
```

## 12. Переопределение и создание нового

```
\newenvironment{НОВОЕ ИМЯ}[КОЛИЧЕСТВО АРГУМЕНТОВ]{
% Здесь располагаются открывающие команды и/или текст;
% в них (и только здесь) можно использовать #1,...,#9
% для обозначения места куда будут вставлены аргументы.
}{
%
% Здесь располагаются закрывающие команды и/или текст;
%
}

\renewenvironment{СТАРОЕ ИМЯ}[КОЛИЧЕСТВО АРГУМЕНТОВ]{...}{...}

\def\КОМАНДА<параметры>{ТЕКСТ} -- примитив TeX для
определения макроса, без проверки существования имени,
где:
КОМАНДА - имя определяемого макроса;
<параметры> - необязательный аргумент типа #1#2...#9 ;
ТЕКСТ - замещающие команды и/или текст;

\newcommand{\НОВОЕ ИМЯ}[КОЛИЧЕСТВО АРГУМЕНТОВ]{
% Здесь располагаются замещающие команды и/или текст;
% в них можно использовать #1,...,#9
% для обозначения места куда будут вставлены аргументы.
}

\renewcommand{\СТАРОЕ ИМЯ}[КОЛИЧЕСТВО АРГУМЕНТОВ]{...}
```

---

## 16. Препринты

Препринт -- это некоторый документ, отпечатанный в виде уменьшенной (в два раза) книжечки с двусторонней печатью.

**Сделать препринт очень просто** следующим образом:

- а) готовим документ в TeX-е: `fname.tex`
- б) преобразуем его в Post Script: `fname.ps`
- в) выполняем команду: `"preprint fname.ps"` (нужно поискать эту утилитку);

И в результате получаем файл `fname.ps.prep` (также в формате Post Script), который остается только распечатать на принтере.

Заметим, что в TeX-е можно подготовить текст, полностью свободный от каких либо команд форматирования. Для этого есть окружение `{verbatim}`.

Поэтому, если у вас есть некоторый документ в самом обыкновенном текстовом виде (**ASCII**), его также легко оформить в виде препринта. Конечно, несколько TeX-овских команд придется включить в начало и конец файла.

Ниже показано как это сделать (несколько необязательных команд улучшают расположение на странице):

```
\documentclass[a4paper,twosides]{article}
\usepackage{russian}

\oddsidemargin=0pt
\evensidemargin=0pt
\textwidth=18cm
\topmargin=-2.5cm
\textheight=26cm

\begin{document}
\begin{verbatim}
|-----|
| Простой текст |
|-----|
\end{verbatim}
\end{document}
```

---

*Геннадий М.*

Last modified: Tue Aug 3 16:08:29 EDT 1999