

Введение в \LaTeX

Занятие 1

Даниил Дрябин

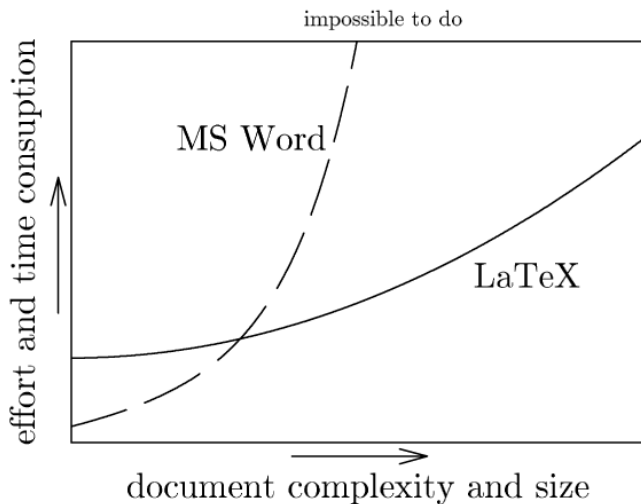
Студсовет ФПМИ

весна 2022

- 1 Почему \LaTeX ?
- 2 Основы синтаксиса
- 3 Набор формул
- 4 Форматирование текста
- 5 Простейшие окружения
- 6 Заключительные указания

Почему \LaTeX ?

Почему \LaTeX ?



Оснoвы синтаксиса

Наш первый документ

Простейший документ, написанный в теке, выглядит так:

```
\documentclass[a4paper, 12pt]{article}

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

Переходим на русский язык

Наше следующее естественное желание — сделать то же самое на русском языке:

```
\documentclass[a4paper, 12pt]{article}

\begin{document}
    Привет, мир!
\end{document}
```

Однако код выше **не скомпилируется!**

Правильно переходим на русский язык

Нам нужно уточнить, что в документе будет русский язык, чтобы система работала с ним корректно, а также указать кодировки. Для надежности, и кодировку документа, и кодировку файла с исходным кодом.

```
\documentclass[a4paper, 12pt]{article}

\usepackage[english, russian]{babel}
\usepackage[T2A]{fontenc}
\usepackage[utf8]{inputenc}

\begin{document}
    Привет, мир!
\end{document}
```

Теперь все работает!

Преамбула

Строка вида `\usepackage{name}` — это подключение некоторого пакета. Нам часто будут пригождаться разные пакеты, и их подключение **всегда** следует выносить в **преамбулу**. В настоящих проектах хорошим тоном является выделять преамбулу в отдельный файл. В качестве спойлера, делается это так:

```
\input{preamble} % Здесь спрятана подготовительная информация

\begin{document}
  % Здесь пишется сам текст
\end{document}
```

Подробно о структурировании больших проектов мы поговорим потом.

Простейшие формулы

Переходим к главному — математике. Математические выкладки в теке можно писать в двух режимах. Формулы посреди строки заключаются в знаки доллара:

Читателю предлагается проверить, что выполнено равенство $2 + 2 = 4$.

Формулы на новой строке и по центру заключаются экранированные квадратные скобки:

Читателю предлагается проверить, что выполнено следующее равенство:

$$2 + 2 = 4$$

Второй вариант набора центрированных формул имеет вид
$$2 + 2 = 4$$
. Однако он считается устаревшим и не рекомендуется к использованию.

Простейшие формулы

Хорошо, мы понимаем как набирать элементарные вещи, но как быть с тем, чего нет на клавиатуре? Тех работает с **командами**, имеющими, например, такой вид: `\command`, `\command{arg}`, `\command[opt_arg]{arg1}{arg2}`. Этих команд — тысячи, но частые запоминаются за пару дней практики, а остальные легко гуглятся. Советуем, например, [этот сайт](#).

Рассмотрим такой пример:

```
\[ \int_{-\infty}^{+\infty} e^{-\frac{x^2}{2}} = \sqrt{2 \pi} \]
```

Этот код дает следующий результат:

$$\int_{-\infty}^{+\infty} e^{-\frac{x^2}{2}} = \sqrt{2\pi}$$

Внимание стоит обратить на команды и индексы! Далее мы это повторим.

Набор формул

Индексы

Верхние и нижние индексы создаются следующим образом:

```
\[x_n, x^k, x_n^k, x^k_n, x_{i + j}^{2022}\]
```

Этот код дает следующий результат:

$$x_n, x^k, x_n^k, x^k_n, x_{i+j}^{2022}$$

Греческие буквы

Получить буквы греческого алфавита в теке очень просто:

```
\[\alpha, \beta, \Gamma, \Delta, \Omega, \epsilon, \xi, \phi\]
```

Этот код дает следующий результат:

$$\alpha, \beta, \Gamma, \Delta, \Omega, \epsilon, \xi, \phi$$

Кстати, как думаете, почему не работает команда `\Alpha`? Отметим еще, что пока некоторые из букв выглядят немного непривычно для русскоязычного читателя. Мы исправим это, но потом. Многие другие особые символы тоже можно получить с помощью команд, например, кванторы `\forall` (\forall) и `\exists` (\exists).

Бинарные операции

Еще раз вспомним, как работать с дробями:

```
\[\frac{\frac{1}{2} + \frac{3}{4}}{\frac{1}{8}} =
```

```
\frac{5}{4} \cdot 8 = 10\]
```

Этот код дает следующий результат:

$$\frac{\frac{1}{2} + \frac{3}{4}}{\frac{1}{8}} = \frac{5}{4} \cdot 8 = 10$$

В процессе нам встретился символ умножения, получаемый командой `\cdot`. Да, использовать обычные символы в качестве бинарных операций — это моветон, если речь идет о чем-то более сложном, чем плюс или минус. Вот несколько других примеров команд для бинарных операций:

`\times` (\times), `\ge` (\geq), `\cong` (\cong), `\oplus` (\oplus), `\sim` (\sim)

Операторы

Некоторые команды — **операторы**, не только создают на печати набор символов нужного вида, но и проставляют отступ для аргумента этого оператора. Сравним три способа написать $\sin x$:

```
$\sin x$, $\mathrm{\sin}\{x\}$, $\sin{x}$
```

Этот код дает следующий результат:

```
 $\sin x$ ,  $\sin x$ ,  $\sin x$ 
```

Ясно, что предпочтительным является третий способ. Бывает, конечно, и так, что по умолчанию в теке нет оператора с нужным нам набором букв, например, rk , Tor или Quot . Тогда можно создавать свои операторы. Как это делать — мы обсудим позднее.

Скобки

В теке есть все привычные нам виды скобок, причем в самых разных вариациях:

```
\[(\frac{\pi}{2}),  
\big( \frac{\pi}{2} \big),  
\bigg( \frac{\pi}{2} \bigg),  
\left( \frac{\pi}{2} \right),  
\left\{ \frac{\pi}{2} \right\},  
\left[ \frac{\pi}{2} \right],  
\left\lfloor \frac{\pi}{2} \right\rfloor
```

Этот код дает следующий результат:

$$\left(\frac{\pi}{2}\right), \left(\frac{\pi}{2}\right), \left(\frac{\pi}{2}\right), \left(\frac{\pi}{2}\right), \left\{\frac{\pi}{2}\right\}, \left[\frac{\pi}{2}\right], \left\lfloor\frac{\pi}{2}\right\rfloor$$

Подводные камни

Как мы уже заметили, ряд символов, которые бывают нужны нам в тексте, имеют особое смысловое значение в теке, и чтобы лишить их этого значения, их следует **экранировать**.

```
\$, \backslash, \{, \}, \&, \#, \_, \%
```

Этот код дает следующий результат:

```
$, \, {, }, &, #, _, %
```

Форматирование текста

Структуризация документа

Мы научились набирать текст и формулы, но когда текста становится много, возникает естественное желание разбивать его на небольшие блоки. Для этого в теке есть команды `\section{name}`, `\subsection{name}` и даже `\subsubsection{name}`. Для особенно больших проектов есть и другие команды, но нам они не потребуются.

```
\section{Мой раздел}
\subsection{Мой подраздел}
\dots % Обычно, двух уровней вложенности хватает.

\subsection*{Подраздел, но без номера}
\dots % Такие команды можно использовать как подзаголовки.

\subsection{Еще подраздел с номером}
\dots % Нумерация не нарушится!
```

Стиль текста

Еще одно естественное желание — это выбор стиля текста. В теке есть два подхода к выбору стиля.

- Использование команд:

```
\textbf{Жирный}, \textit{курсивный}, обычный.
```

Этот код дает следующий результат:

Жирный, *курсивный*, обычный.

- Использование **областей видимости** и **модификаторов**:

```
{\bfseries \itshape Здесь выделенное}, а здесь нет.
```

Этот код дает следующий результат:

Здесь выделенное, а здесь нет.

Размер и шрифт

Изменение размера шрифта производится с помощью модификаторов. Их довольно много, приведем несколько примеров:

`{\Large Большой текст}`, текст поменьше, `{\small вообще маленький текст жуть}`.

Этот код дает следующий результат:

Большой текст, текст поменьше, вообще маленький текст жуть.

Заменять шрифт на один из стандартных вариантов можно как модификатором, так и командой:

Модификатор	Команда	Результат
<code>\rmfamily</code>	<code>\textrm{}</code>	Шрифт
<code>\sffamily</code>	<code>\textsf{}</code>	Шрифт
<code>\ttfamily</code>	<code>\texttt{}</code>	Шрифт
<code>\scshape</code>	<code>\textsc{}</code>	ШРИФТ

Простейшие окружения

Что такое окружение?

Окружение — это способ инкапсулировать много модификаторов стиля в единый блок, в который потом можно будет заворачивать текст.

Использование окружения имеет вид

```
\begin{environment}...\end{environment}.
```

```
\begin{center}
```

Этот текст будет центрирован.

```
\end{center}
```

```
\begin{flushright}
```

А этот смещен вправо.

```
\end{flushright}
```

Этот код дает следующий результат:

Этот текст будет центрирован.

А этот смещен вправо.

Окружения-теоремы

Особый случай окружения — это **теорема**. Окружение такого вида предваряет текст внутри надписью вида «Теорема N» с согласованной нумерацией. Сегодня ограничимся простейшим примером.

```
\usepackage{amsthm}
\theoremstyle{plain}
\newtheorem{theorem}{Теорема}
\newtheorem{lemma}{Лемма}

\begin{lemma} 1 + 1 = 3. \end{lemma}

\begin{theorem} 2 + 2 = 5. \end{theorem}
```

Этот код дает следующий результат:

Лемма 1. $1 + 1 = 3$.

Теорема 1. $2 + 2 = 5$.

О создании своих теорем, и вообще своих окружений мы поговорим позднее.

Заключительные указания

Заклучительные указания

Перед тем, как приступить к практике, вам стоит обратить внимание еще на несколько вещей.

- Пакеты от Американского Математического Общества, подключение которых никогда не будет лишним:

```
\usepackage{amsmath, amsfonts, amssymb, amsthm, mathtools}
```

- В теке масса видов тире и дефисов. Пользоваться ими лучше так:
 - "--- для разделения подлежащего и сказуемого
 - --- для других ситуаций с тире в тексте
 - -- для указания числовых промежутков в тексте
 - - для дефисов в тексте
 - \$-\$ для знака минус
- В теке есть свои, правильные многоточия — `\dots` и его вариации, с которыми мы еще столкнемся. А также свои кавычки — `<<` и `>>`.

Всё!

A large, stylized 3D graphic of the Russian word 'ВСЁ!' (Everything!) in yellow with red outlines and shadows, set against a solid blue background. The letters are bold and blocky, with a slight 3D effect. The exclamation mark is also stylized, with a yellow dot and a red shadow.