

Введение в \LaTeX

Занятие 2

Даниил Дрябин

Студсовет ФПМИ

весна 2022

- 1 Оформление документа
- 2 Изображения и таблицы
- 3 Создание команд
- 4 Окружения
- 5 Набор формул II

Оформление документа

Основные разделы

Вспомним основные разделы, в порядке убывания уровня значимости:

- `part` — часть
- `chapter` — глава
- `section` — раздел
- `subsection` — подраздел
- `subsubsection` — подподраздел
- `paragraph` — абзац
- `subparagraph` — подабзац

Синтаксис имеет такой вид: `\section_name[short title]{full title}`.

Не все разделы доступны для каждого класса документа, лучше смотреть информацию о каждом отдельно. Для нас будет актуален в основном класс `article`, не поддерживающий только команду `\chapter`.

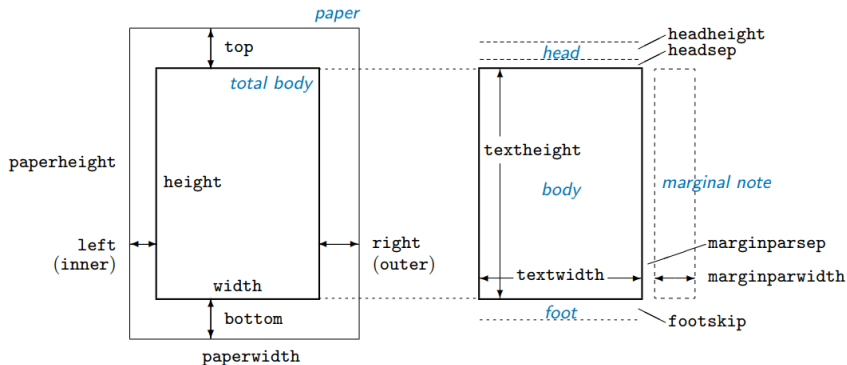
Генерация разделов

- `\maketitle` — генерация заголовка, использует значения параметров, задаваемых командами `\author{}`, `\date{}` и `\title{}`
- `\tableofcontents` — генерация содержания, использует все разделы с нумерацией
- `\listoffigures` — генерация списка иллюстраций
- `\listoftables` — генерация списка таблиц
- `\appendix` — разделы ниже такой команды будут считаться разделами приложения, со своими правилами нумерации

Все эти автоматизированные блоки можно настраивать на свое усмотрение. Мы будем еще будем говорить о некоторых из них позднее.

Пакет geometry

Пакет `geometry` позволяет гибко настраивать числовые параметры, характеризующие поля и колонтитулы страниц.



Установка числовых параметров

Поддерживаемые единицы измерения:

- pt, ex (высота буквы x), em (ширина буквы M)
- mm, cm, in

Параметры пакета `geometry` устанавливаются так:

```
\geometry{top=25mm}  
\geometry{bottom=30mm}  
\geometry{left=20mm}  
\geometry{right=20mm}
```

Есть и другие параметры:

```
\linespread{1}  
\setlength{\parindent}{20pt}  
\setlength{\parskip}{12pt}
```

Колонтитулы

Удобный способ задавать колонтитулы — пакет `titleps`:

```
\usepackage{titleps}

\newpagestyle{main}{
  \setheadrule{0.4pt}
  \sethead{лево}{центр}{право}
  \setfootrule{0.4pt}
  \setfoot{лево}{центр}{право}
}

\pagestyle{main}
```

Команда `\pagestyle{name}` — это тоже модификатор! Если требуется наложить эффект только на текущую страницу, используется команда `\thispagestyle{name}`.

Пакет soul

Пакет soul предоставляет несколько команд для модификации начертания:

```
\usepackage{soulutf8}  
  
\so{letterspacing}  
\caps{Capitals}  
\ul{underlining}  
\st{overstriking}
```

Этот код дает следующий результат:

```
l e t t e r s p a c i n g  
C A P I T A L S  
u n d e r l i n i n g  
o v e r s t r i k i n g
```

Отступы

В теке можно по-разному регулировать отступы:

- `\hspace{length}`, `\vspace{length}` — пробел по горизонтали и вертикали (вертикальный пробел имеет эффект по окончании текущей строки)
- `\hspace*{length}`, `\vspace*{length}` — аналоги пробелов, не сокращаемые концом строки
- `\hfill`, `\vfill` — «распорки», гарантирующие, что контент до и после них расположен в противоположных концах страницы

Переходы на новую строку

В теке можно по-разному регулировать переходы на новую строку:

- Пустая строка, как и команда `\par`, совершают переход на новый абзац, начинающийся с красной строки (эти операции *идемпотентны*)
- Команды `\\` и `\newline` совершают насильственный переход на новую строку (аналог для страниц — `\newpage`)
- Команда `\linebreak` совершает переход на новую строку, не нарушая правил форматирования текста (аналог для страниц — `\pagebreak`)

Чтобы иметь красную строку в первом абзаце каждого раздела, как положено в русскоязычной типографии, следует использовать команду `\usepackage{indentfirst}`.

Изображения и таблицы

Создание таблиц

Создание таблиц в теке имеет следующий синтаксис:

Рассмотрим такую таблицу:

```
\begin{tabular}{||c|c|||r|l||}
  1 & 2 & aligned & aligned \\
\hline
  3 & 4 & to the right & to the left \\
\end{tabular}
```

Этот код дает следующий результат:

Рассмотрим такую таблицу:

| | | | |
|-----|-----|--------------|-------------|
| 1 | x | aligned | aligned |
| y | 1 | to the right | to the left |

В большинстве случаев таблицы удобно заворачивать в другие окружения с форматированием, например, `center`.

Вставка изображений

Вставка изображений в теке имеет следующий синтаксис:

Посмотрим на злого котика:

```
\includegraphics[width = 0.25\textwidth]{img/cat}
```

Этот код дает следующий результат:

Посмотрим на злого котика:



Как и в случае таблиц, изображения удобно заворачивать в другие окружения с форматированием, например, `center`.

Создание команд

Команды

В теке можно создавать свои команды, чтобы переиспользовать код и упрощать себе жизнь. Типичный пример — команда написания частной производной или символа \mathbb{R} :

```
\newcommand{\deriv}[2]{\frac{\partial #1}{\partial #2}}
\newcommand{\R}{\mathbb R}

\[\forall x, y \in \R: df(x, y) = \deriv{f}{x}(x, y)dx +
\deriv{f}{y}(x, y)dy\]
```

Этот код дает следующий результат:

$$\forall x, y \in \mathbb{R} : df(x, y) = \frac{\partial f}{\partial x}(x, y)dx + \frac{\partial f}{\partial y}(x, y)dy$$

Если выбранное вами название команды уже занято, выберите другое или воспользуйтесь инструментом со следующего слайда.

Переопределение команд

Мы сталкивались с такой проблемой, что некоторые греческие буквы имеют непривычные нам очертания. Это можно исправить так:

```
\renewcommand{\phi}{\varphi}
\renewcommand{\epsilon}{\varepsilon}
```

Теперь мы имеем красивые буквы ϕ и ϵ !

Этот код дает следующий результат:

Теперь мы имеем красивые буквы φ и ε !

Можно переопределять и любые другие команды, в частности, те, которые принимают на вход аргументы. Но лучше прибегать к такому только тогда, когда вы *точно* понимаете, что делаете!

Операторы

Можно создавать свои операторы командой `\DeclareMathOperator`:

```
\DeclareMathOperator{\Kerr}{Ker}
\DeclareMathOperator{\Imm}{Im}

\[\dim \Kerr \phi + \dim \Imm \phi = \dim V\]
```

Этот код дает следующий результат:

$$\dim \operatorname{Ker} \varphi + \dim \operatorname{Im} \varphi = \dim V$$

Операторы с пределами

Вариант команды `\DeclareMathOperator*` создает операторы, верхний и нижний индексы которых являются пределами:

```
\DeclareMathOperator*{\argmax}{argmax}

\[\argmax_{x \in \mathbb{R}} f(x) = +\infty\]
```

Этот код дает следующий результат:

$$\argmax_{x \in \mathbb{R}} f(x) = +\infty$$

Подробнее о механизме работы со «звездочкой» мы поговорим позднее.

Бинарные операции

Очень редко, но все же возникает необходимость создавать свои бинарные операции, которых нет среди стандартных операций. Для этого используется команда `\mathbin`:

```
\newcommand{\percent}{\mathbin{\%}}
```

```
\[A \percent B\]
```

Этот код дает следующий результат:

$$a \% b$$

Окружения

Нумерованные и маркированные списки

Списки в теке создаются следующим образом:

```
\begin{itemize}
  \item Первый пункт
  \item Второй пункт
\end{itemize}

\begin{enumerate}
  \item Первый пункт с номером
  \item Второй пункт с номером
\end{enumerate}
```

Этот код дает следующий результат:

- Первый пункт
- Второй пункт
- ❶ Первый пункт с номером
- ❷ Второй пункт с номером

Вложенные списки

Списки разного типа можно вкладывать друг в друга, при этом символы-маркеры и стиль нумерации будут меняться, чтобы отличаться от стиля списков выше. Можно управлять тем, как именно будут меняться стили, но об этом мы поговорим позднее.

Отметим еще, что выбирать символ-маркер можно самому для каждого пункта в отдельности:

Докажем, что $a = b \Leftrightarrow b = a$ для любых $a, b \in \mathbb{R}$.

```
\begin{itemize}
```

```
\item[ $\Leftarrow$ ] Очевидно.
```

```
\item[ $\Rightarrow$ ] Тривиально.
```

```
\end{itemize}
```

Этот код дает следующий результат:

Докажем, что $a = b \Leftrightarrow b = a$ для любых $a, b \in \mathbb{R}$.

\Leftarrow Очевидно.

\Rightarrow Тривиально.

Создание теорем

Напомним, что интерфейс работы с теоремами обеспечивается пакетом `amsthm`. Встроенные стили теорем — это `plain`, `definition` и `remark`. Теоремы могут быть нумерованными, причем нумерацию можно согласовать с нумерацией разделов (или любым другим счетчиком), и ненумерованными.

```
\theoremstyle{plain}
\newtheorem{theorem}{Теорема}[section]
\newtheorem{corollary}{Следствие}[theorem]
\newtheorem*{definition}{Определение}

\section{Первый раздел}

\begin{theorem}Текст.\end{theorem}
\begin{theorem}Еще текст.\end{theorem}
\begin{proof}Тривиально.\end{proof}
\begin{corollary}И еще текст.\end{corollary}
\begin{definition}Что-то новое.\end{definition}
```


Создание теорем

Код с предыдущего слайда дает следующий результат:

1 Первый раздел

Теорема 1.1. *Текст.*

Теорема 1.2. *Еще текст.*

Доказательство. Тривиально. □

Следствие 1.2.1. *И еще текст.*

Определение. *Что-то новое.*

О создании своих стилей теорем, переопределении среды «доказательства» и о счетчиках в нумерации мы поговорим позднее.

Набор формул II

Математические шрифты

В теке есть большое разнообразие математических шрифтов. Изменение шрифта в математическом режиме делается специальными командами:

| Команда | Пример | Результат |
|-----------------------|--|----------------------------|
| <code>\mathbb</code> | <code>\$x \in \mathbb R\$</code> | $x \in \mathbb{R}$ |
| <code>\mathbf</code> | <code> \$\mathbf{Ax} = \mathbf{b}\$</code> | $\mathbf{Ax} = \mathbf{b}$ |
| <code>\mathrm</code> | <code> \$\phi \in \mathrm{GL}(V)\$</code> | $\phi \in \mathrm{GL}(V)$ |
| <code>\mathsf</code> | <code> \$\mathsf{E}\xi < +\infty\$</code> | $E\xi < +\infty$ |
| <code>\mathcal</code> | <code> \$\psi \in \mathcal{L}(V)\$</code> | $\psi \in \mathcal{L}(V)$ |

Есть и другие шрифты, как встроенные, так и пользовательские, но пользовательские шрифты становятся доступны только при подключении соответствующих пакетов.

Слишком длинные команды мы можем сокращать, создавая свои команды с тем же эффектом.

Пределы

Для некоторых команд, таких как `\sum` и `\prod`, верхний и нижний индексы по возможности размещаются над текстом, генерируемым командой.

Контролировать поведение индексов можно командой `\limits`.

Внутри строки: $e^x = \sum_{n=0}^{+\infty} \frac{x^n}{n!} = \sum\limits_{n=0}^{+\infty} \frac{x^n}{n!}$.

На своей строке:

$$[e^x = \sum_{n=0}^{+\infty} \frac{x^n}{n!} = \sum\limits_{n=0}^{+\infty} \frac{x^n}{n!}]$$

Этот код дает следующий результат:

Внутри строки: $e^x = \sum_{n=0}^{+\infty} \frac{x^n}{n!} = \sum_{n=0}^{+\infty} \frac{x^n}{n!}$.

На своей строке:

$$e^x = \sum_{n=0}^{+\infty} \frac{x^n}{n!} = \sum_{n=0}^{+\infty} \frac{x^n}{n!}$$

Стрелки с надписями

Стрелки с текстом тоже можно создавать по-разному:

```
\[\phi \leftrightharpoonup_e^{\text{просто текст}} A \in M_n(F)\]
```

```
\[\phi \xleftrightharpoonup[\text{просто текст}]{e} A \in M_n(F)\]
```

Этот код дает следующий результат:

$$\varphi \leftrightarrow_e^{\text{просто текст}} A \in M_n(F)$$

$$\varphi \xleftarrow[\text{просто текст}]{e} A \in M_n(F)$$

Отметим, что использование команды `\limits` и длинных стрелок не рекомендуется для внутристрочных формул.

Знаки над символами

В теке есть команды, позволяющие рисовать что-то над символами. Приведем несколько полезных примеров:

| Короткий вариант | Результат | Длинный вариант | Результат |
|------------------------------|----------------|--------------------------------------|-----------------------|
| <code>\$\dot a\$</code> | \dot{a} | — | — |
| <code>\$\ddot a\$</code> | \ddot{a} | — | — |
| <code>\$\mathring a\$</code> | \mathring{a} | — | — |
| <code>\$\hat a\$</code> | \hat{a} | <code>\$\widehat{AB}\$</code> | \widehat{AB} |
| <code>\$\tilde a\$</code> | \tilde{a} | <code>\$\widetilde{AB}\$</code> | \widetilde{AB} |
| <code>\$\bar a\$</code> | \bar{a} | <code>\$\overline{AB}\$</code> | \overline{AB} |
| <code>\$\vec a\$</code> | \vec{a} | <code>\$\overrightarrow{AB}\$</code> | \overrightarrow{AB} |

Опять же, слишком длинные команды можно сокращать!

Полезные окружения

При наборе формул бывают нужны следующие нумерованные и ненумерованные окружения:

- `equation` и `equation*`
- `align` и `align*`
- `multline` и `multline*`
- `aligned`
- `cases`

Подробнее о нумерации и ссылках на уравнения мы будем говорить позднее.

Всё!



ВСЁ!