Введение в LaTeX

Занятие 2

Даниил Дрябин

Студсовет ФПМИ

осень 2022

Вопросы с прошлого занятия

Макросы в VS Code

Во многих редакторах, поддерживающих латех, есть возможность упрощать себе жизнь, задавая свои сочетания клавиш. В VS Code это делается так: проходим в Preferences > Keyboard shortcuts и выбираем Open Keyboard Shortcuts. В открывшемся JSON-файле указываем свои команды в следующем виде:

```
{
    "key": "cmd+shift+9",
    "command": "editor.action.insertSnippet",
    "when": "resourceExtname == .tex",
    "args": {
        "snippet": "\\left(${TM_SELECTED_TEXT}\\right)$0"
    }
}
```

Теперь сочетание клавиш cmd+shift+9 будет оборачивать выделенный текст в скобки `\left(..\right)` и располагать курсор в позиции $0, то есть после скобок.

Кириллица в математическом режиме

Для того, чтобы использовать кириллицу (и вообще символы Юникода) в математическом режиме, требуется подключение специального пакета `mathtext`.

Альтернативный вариант — использовать команду `\text{}` из пакета `amsmath`. Существенной разницы между этим способом и предыдущим мне неизвестно.

```
$\text{А} + \text{Б} = \text{В}$
```

Этот код дает следующий результат:

$$\text{А} + \text{Б} = \text{В}$$

Оформление документа

Отступы

В texe 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 техе 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}`.

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`.

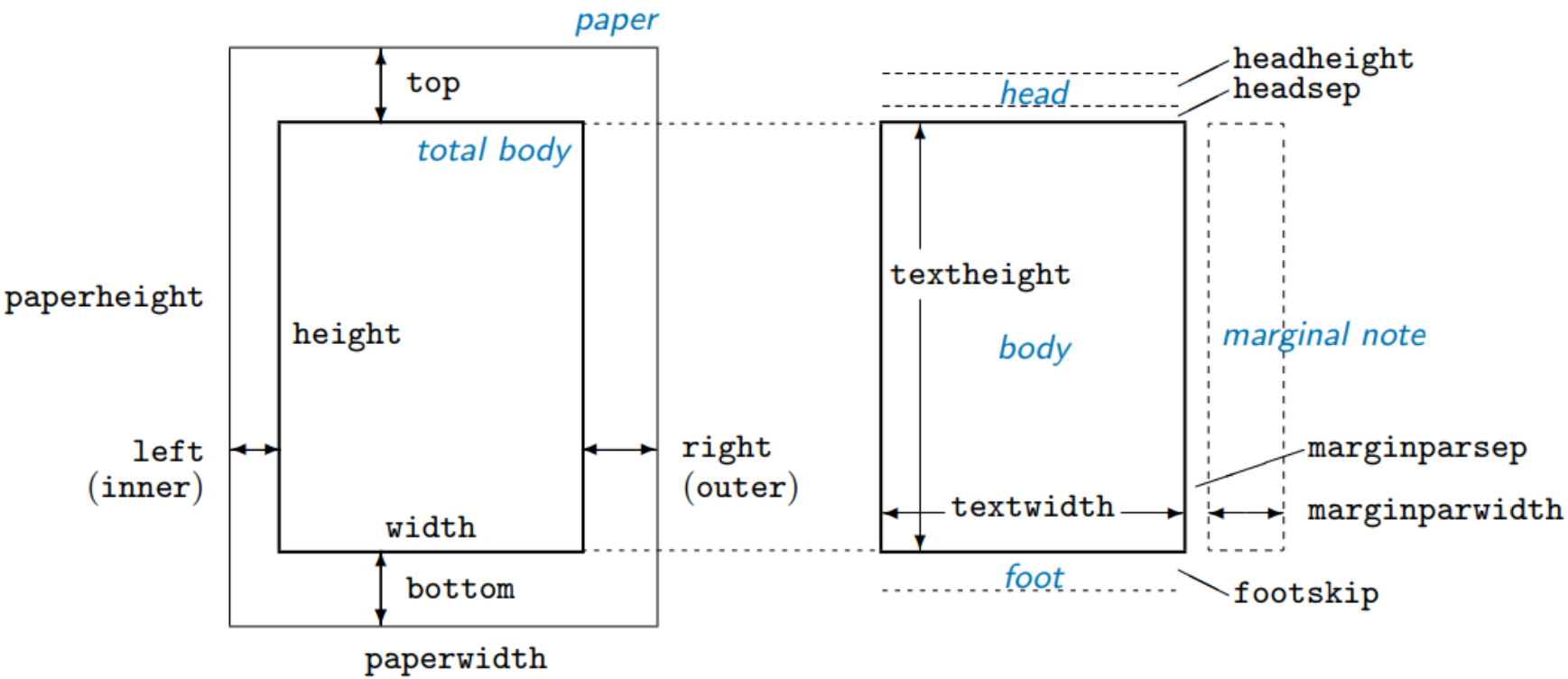
Генерация разделов

- `\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 будет 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
CAPITALS
underlining
~~overstriking~~

Изображения и таблицы

Создание таблиц

Создание таблиц в texe имеет следующий синтаксис:

```
Рассмотрим такую таблицу:

\begin{tabular}{||c|c|||r|l||}
    1 & x & aligned & aligned \\
    \hline
    y & x & 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 texe 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 texe 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 $\phi$ и $\epsilon$!
```

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:

$$\operatorname*{argmax}_{x \in \mathbb{R}} f(x) = +\infty$$

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 texe 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

1. Первый пункт с номером
2. Второй пункт с номером

Маркеры в списках

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

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

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

На своей строке:
\[e^x = \sum_{n=0}^{+\infty}\frac{x^n}{n!} =
\sum\limits_{n=0}^{+\infty}\dfrac{x^n}{n!}\]
```

Этот код дает следующий результат:

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

На своей строке:

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

Стрелки с надписями

Стрелки с текстом тоже можно создавать по-разному:

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

Этот код дает следующий результат:

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

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

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

Знаки над символами

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

Короткий вариант	Результат	Длинный вариант	Результат
`$\dot a$`	$\dot{a}$	—	—
`$\ddot a$`	$\ddot{a}$	—	—
`$\mathring a$`	$\mathring{a}$	—	—
`$\hat a$`	$\hat{a}$	`$\widehat{AB}$`	$\widehat{AB}$
`$\tilde a$`	$\tilde{a}$	`$\widetilde{AB}$`	$\widetilde{AB}$
`$\bar a$`	$\bar{a}$	`$\overline{AB}$`	$\overline{AB}$
`$\vec a$`	$\vec{a}$	`$\overrightarrow{AB}$`	$\overrightarrow{AB}$

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!