

Введение в \LaTeX

Занятие 3

Даниил Дрябин

Студсовет ФПМИ

осень 2022

- 1 Уточнения к прошлому занятию
- 2 Набор формул III
- 3 Ссылки
- 4 Счётчики
- 5 Создание окружений

Уточнения к прошлому занятию

Разица в переходах на новую строку

- Команда `\par` эквивалентна пустой строке (двум символам `\n`), но при этом делает код менее читаемым
- Команда `\newline` совершает *насильственный* переход на новую строку в том месте, в котором она указана (в обычном текстовом режиме то же самое делает команда `\\`), при этом важно, чтобы в текущей строке был какой-то ещё контент
- Команда `\linebreak` даёт латеху *рекомендацию* завершить строку, когда он посчитает возможным, при этом завершённая строка выравнивается по текущим правилам (например, по ширине)
- Ни одна из команд `\newline`, `\\`, `\linebreak` не начинает нового абзаца
- Команды `\newpage` и `\pagebreak` устроены так же, как их аналоги для строк, описанные выше

Ни одна команда с этого слайда не рекомендуется к использованию!

Пример использования окружения cases

Окружение cases — типичный пример окружения, в котором & и \\ приобретают особое значение.

Определим функцию Дирихле:

```
\[
  D(x) := \begin{cases}
    1 & x \in \mathbb{Q} \\
    \\
    0 & x \notin \mathbb{Q}
  \end{cases}
\]
```

Этот код дает следующий результат:

Определим функцию Дирихле:

$$D(x) := \begin{cases} 1 & x \in \mathbb{Q} \\ 0 & x \notin \mathbb{Q} \end{cases}$$

Набор формул III

Окружение equation

При наборе выключных формул раньше мы использовали `\[\]`. А если нам хочется несколько строк в формуле, или нумерацию формул? Для нумерации формул можно использовать, например, окружение `equation`.

Коммутативность:

```
\begin{equation}
  xy = yx.
\end{equation}
```

Коммутативность:

$$xy = yx. \tag{1}$$

Нумерация формул в `article` привязана к секциям, однако это всё можно настроить под себя.

Окружение align

У `equation` есть множество недостатков, из самых очевидных — невозможность перехода на новую строку. Самый популярный и хороший способ писать выключные формулы — это окружение `align`. По мимо прочего, оно позволяет делать настраиваемое выравнивание.

Коммутативность:

```
\begin{align}
  xy &= yx, \\
  ab &= ba.
\end{align}
```

Коммутативность:

$$xy = yx, \tag{2}$$

$$ab = ba. \tag{3}$$

Другие окружения

Существует множество других, более специфичных окружений для формул, например, такие:

- `multline` — многострочное окружение, в котором первая строка выравнивается по левому краю, последняя — по правому, а центры всех промежуточных располагаются равномерно между центрами первой и последней
- `gather` — упрощенный вариант окружения `align` с единственным способом выравнивания — по центру
- `split` — вариант окружения `align`, в котором нумеруются не строки по отдельности, а весь их набор как один объект

Использование окружения со «звездочкой» (например, `align*`) отключает нумерацию строк в нем.

Управление нумерацией

Можно по-своему пометить конкретную формулу с помощью `\tag`.

```
\[
  F = \frac{dp}{dt} \tag{III 3H}
\]
```

Этот код дает следующий результат:

$$F = \frac{dp}{dt} \tag{III 3H}$$

Вложенные окружения

Внутри формул есть свои окружения. Например, для набора системы уравнений используется `cases`.

```
\[
  \begin{cases}
    ax + by = e, \\
    cx + dy = f.
  \end{cases}
\]
```

Этот код дает следующий результат:

$$\begin{cases} ax + by = e, \\ cx + dy = f \end{cases}$$

Другой пример — окружение `aligned`, аналогичное окружению `align`, но используемое в качестве вложенного окружения.

Окружения матриц

Еще один полезный пример вложенного окружения — матрицы. Существует обычное окружение `matrix`, работающее как таблица без линий-разделителей, и его вариации, такие как `vmatrix` и `pmatrix`, определяющие скобки вокруг матрицы.

```
\[  
  \begin{pmatrix}  
    a & b \\  
    c & d  
  \end{pmatrix}  
\]
```

Этот код дает следующий результат:

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

Большие матрицы

В теке есть несколько специальных видов точек для больших матриц:

```
\[
\begin{pmatrix}
a_{11} & a_{12} & \cdots & a_{1n} \\
a_{21} & a_{22} & \cdots & a_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
a_{n1} & a_{n2} & \cdots & a_{nn}
\end{pmatrix}
\]
```

Этот код дает следующий результат:

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}$$

Пакет nicematrix

В специально разработанном пакете `nicematrix` есть множество оберток над матрицами. Например, в нем удобно делать такие расширенные матрицы:

```
\usepackage{nicematrix}
\[
  \begin{pNiceArray}{cc|c}
    a_{11} & a_{12} & b_1 \\
    a_{21} & a_{22} & b_2
  \end{pNiceArray}
\]
```

Этот код дает следующий результат:

$$\left(\begin{array}{cc|c} a_{11} & a_{12} & b_1 \\ a_{21} & a_{22} & b_2 \end{array} \right)$$

Так же там удобно делать, например, блочные матрицы. Функционал большой, и имеется приятный [мануал с иллюстрациями](#).

Ссылки

Internal

Можно размещать внутренние ссылки на формулы, например, на всем известный III закон Ньютона (III ЗН) на слайде 10. Чтобы такое работало, код должен выглядеть так:

```
\[
  \label{eq:3n1}
  F = \frac{dp}{dt} \tag{III ЗН}
\]
```

Для ссылки на формулу и страницу я использовал `\eqref{eq:3n1}` и `\pageref{eq:3n1}`, соответственно. Такие ссылки можно делать на любой объект, помеченный командой `label`. По умолчанию, для обращения к лейблам используется команда `\ref`, однако существуют и более специфичные команды, такие как, например, `\figref` и `\eqref`.

External

Можно делать ссылки на внешние сайты:

Для решения вопроса рекомендуем обратиться на сайт
`\href{http://google.com/}{Google}`.

Этот код дает следующий результат:

Для решения вопроса рекомендуем обратиться на сайт [Google](http://google.com/).

Footnotes¹

Сноски в документах можно делать с помощью команд вида `\footnote{сноска с английского \label{footnote}}`.

На сноски, как и на формулы, можно ссылаться, если предварительно пометить их лейблом: см. сноску [1](#) (`\ref{footnote}`).

¹сноска, с английского

Счётчики

Новый счётчик

Счётчик — это числовая переменная, которую можно использовать в своих целях. Создать счётчик с нулевым значением в теке можно так:

```
\newcounter{name}
```

Можно также создать пару счётчиков, в которой второй счетик будет обнуляться каждый раз, когда инкрементируется первый:

```
\newcounter{name}[other_name]
```

Работа со счётчиками

Значение счётчика можно инкрементировать и задавать непосредственно:

```
\stepcounter{name}  
\addtocounter{name}{number}  
\setcounter{name}{number}
```

Обращение к значению счётчика

К значению счётчика можно обращаться в разном виде:

```
\newcounter{name}
\setcounter{name}{22}

\[\thename, \arabic{name}\]
\[\roman{name}, \Roman{name}\] % positive only
\[\alph{name}, \Alph{name}\] % from 0 to 26 only
```

Этот код дает следующий результат:

22, 22
xxii, XXII
v, V

Стандартные счётчики

В теке есть много встроенных счётчиков, отслеживающих числовые параметры текущего положения в тексте:

- `part`, `section`, etc.
- `page`, `figure`, `table`, `footnote`
- `enumi`, `enumii`, `enumiii`, `enumiv`
- `equation`

Использование счётчиков в списках

Для манипуляций со нумерацией списков используется пакет `enumitem`.

```
\usepackage{enumitem}

\begin{enumerate}[label=\Alph*]
  \item item A
  \setcounter{enumi}{5}
  \item another item
\end{enumerate}
```

Этот код дает следующий результат:

```
A item A
F another item
```


Переопределение маркеров в списках

За значки маркеров в списках отвечают команды вида `\labelitemX`, где вместо X пишется уroveň вложенности списка. Их можно переопределять следующим образом:

```
\renewcommand{\labelitemi}{\circ$}  
\renewcommand{\labelitemii}{\circ$}  
\renewcommand{\labelitemiii}{\circ$}  
\renewcommand{\labelitemiv}{\circ$}
```

Теперь маркеры в списках первых четырёх уровней будут иметь вид \circ .

Создание окружений

Свои окружения

Создание окружений в теке имеет следующий синтаксис:

```
\newenvironment{name}{preceding_code}{succeeding_code}
```

Рассмотрим такой пример:

```
\newenvironment{centerbold}  
{\begin{center}\bfseries}  
{\end{center}}  
  
\begin{centerbold}Текст.\end{centerbold}
```

Этот код дает следующий результат:

Текст.

Свои стили теорем

Создание стилей теорем в теке имеет следующий синтаксис:

```
\newtheoremstyle{stylename} % название стиля
  {spaceabove} % отступ до теоремы
  {spacebelow} % отступ после теоремы
  {bodyfont} % шрифт
  {indent} % отступ
  {headfont} % шрифт заголовка
  {headpunctuation} % отделение заголовка от теоремы
  {headspace} % отступ от заголовка от теоремы
  {headspec} % фиксация самого заголовка
```

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.