

# Инструкция по написанию конспектов к видео лекциям

Посов Илья

4 августа 2020 г.

## 1 Введение

Репозиторий <https://github.com/ipo-kio/pdf-with-video> содержит инструменты для написания конспектов к видео-лекциям. Внутри репозитория находятся:

1.  $\text{\LaTeX}$ -класс для оформления конспектов лекций;
2. пакет `timestamps` для указания меток времени внутри конспектов;
3. приложение, работающее в браузере, для проверки связи конспекта с видео;
4. сами конспекты лекций.

В дальнейших разделах предполагается, что вы сумели клонировать себе репозиторий. Инструкции по работе с `git` при необходимости будут написаны отдельно.

## 2 Установка

Чтобы писать конспекты в  $\text{\LaTeX}$ , необходимо настроить свой дистрибутив  $\text{\TeX}$ , чтобы он мог найти необходимые классы и пакеты. Напомню, что  $\text{\LaTeX}$ -документы начинаются с команды, указывающей класс документа, в нашем случае используется класс `lecture-notes`, и это указывается следующим образом:

```
\documentclass[russian]{lecture-notes}
```

Простейший способ заставить  $\text{\LaTeX}$  увидеть этот класс — это просто скопировать файл `lecture-notes.cls` в папку с конспектом. Но этот метод нужно использовать только в крайнем случае, потому что класс будет постоянно изменяться, и будет очень неудобно постоянно копировать его новые версии. Правильный способ настроить класс — это добавить каталог с классом

в список каталогов, которые ваш дистрибутив  $\TeX$  просматривает при поиске классов.

В случае с MikTeX процесс описан по ссылке <https://miktex.org/kb/texmf-roots>, смотрите нижний раздел про добавление собственных texmf-директорий. Вам нужно добавить директорию `/src/texmf` из репозитория.

В случае, если вы пользуетесь Linux с дистрибутивом texlive, создайте каталог `/texmf/`, это каталог внутри домашнего каталога. И вставьте в нем символическую ссылку на каталог внутри репозитория:

```
mkdir -p ~/texmf/tex/latex
ln -s репозиторий/src/texmf/tex/latex/video-timestamps \
    ~/texmf/tex/latex/video-timestamps
```

### 3 Написание конспекта

Шаблон конспекта расположен в файле `src/texmf/doc/latex/video-timestamps/lecture-notes.tex`. Редакторов для  $\LaTeX$  много, я рекомендую TeXStudio.

Чтобы сделать свой конспект, скопируйте этот файл и сделайте в нем необходимые исправления. Сам файл содержит разметку, которая требуется для конспекта и примеры часто необходимых возможностей, например, как писать формулы и как вставлять изображения. В файле есть только примеры, более полная информация находится здесь, либо в учебниках  $\LaTeX$ . Я рекомендую учебники на сайте Overleaf и LaTeX wiki book.

Готовые конспекты должны быть расположены в каталоге `src/lectures`, по отдельному каталогу на каждую лекцию. Если курс содержит несколько лекций, значит, нужно создать каталог для курса, а внутри каталоги отдельных лекций.

В репозиторий необходимо добавлять только tex файлы с текстом и изображения. Все остальные типы файлов, которые генерирует TeX, например, log или aux файлы, не должны попадать в репозиторий. Они явно игнорируются в `.gitignore`.

### 4 Изображения в конспекте

Примеры, как вставлять изображения, есть в шаблоне конспекта. Здесь обсудим выбор форматов изображений.

Используйте формат jpg для фотографий. Скриншоты сохраняются в формате png.

#### 4.1 Свои рисунки в конспекте

Если вы делаете свой рисунок, то есть три возможности:

1. растровое изображение в формате png;
2. векторное изображение в формате svg;
3. векторное изображение с помощью пакета `tikz`.

Изображения в формате png лучше избегать. Их труднее редактировать, они теряют качество при масштабировании, и они занимают больше места, особенно, если у них несколько версий в git репозитории.

Пакет `tikz` — это хороший вариант, особенно хорошо этот формат подходит для описания графов. Но рисовать с его помощью сложно, потому положения точек вводятся координатами, и в языке есть команды, которые надо изучать. Либо нужно найти программы для рисования, которые способны генерировать `tikz` код.

В общем случае рекомендуется использовать формат svg. Для редактирования изображений в этом формате используйте программу Inkscape, это бесплатный многофункциональный редактор векторной графики типа Corel Draw или Adobe Illustrator. Его в любом случае потребуется установить, если вы решите пользоваться svg изображениями в  $\text{\LaTeX}$ -документах, потому что  $\text{\LaTeX}$  использует ее внутри себя для чтения svg.

Чтобы svg изображения вставлялись в конспект, нужны дополнительные действия по настройке  $\text{\LaTeX}$ . Кроме установки Inkscape проследите, чтобы при компиляции конспекта был указан параметр компиляции `--shell-escape`. При компиляции из командной строки пишете:

```
pdflatex --shell-escape my-notes.tex
```

При компиляции в TeXStudio, откройте меню Options, внутри Configure TeXStudio, вкладка Commands, впишите в PDFLaTeX компилятор нужную опцию, получится что-то типа:

```
pdflatex -synctex=1 -interaction=nonstopmode --shell-escape %.tex
```

Эта опция нужна, чтобы разрешить  $\text{\LaTeX}$  запускать внешние программы, в нашем случае Inkscape, в процессе компиляции. По-умолчанию это отключено из соображений безопасности.

## 5 Проверка конспекта в браузере с видео

Раздел пишется...

## 6 Важные правила при оформлении текста в $\text{\LaTeX}$

В этом разделе обсуждаются частые ошибки и трудности, которые возникают при использовании  $\text{\LaTeX}$ .

## 6.1 Разделение представления и содержания

При наборе  $\text{\LaTeX}$  документа важно разделить содержание документа и описание того, как документ выглядит. Это позволяет оформить документ единообразно, т.е. чтобы одинаковые по смыслу элементы выглядели тоже одинаково. Например, все заголовки разделов должны иметь одинаковый шрифт и отступы вокруг. Подписи под изображениями должны быть одинаково выровнены и тоже иметь одинаковый шрифт. Если нужно начинать все главы с новой страницы, это тоже должно быть описано один раз на уровне оформления документа, и об этом не следует каждый раз заботиться при наборе очередной главы. Может добавиться дополнительное требование, что начинать главы нужно всегда с нечетной страницы, этого было бы трудно добиться, расставляя переводы страниц вручную.

Аналогия разделения оформления и содержания в мире веб разработки — это разделение документа на `html`, т.е. содержание, и `css` таблицу стилей, т.е. оформление.

Какие команды относятся к оформлению:

1. команды установки размеров страницы, полей, колонтитулов;
2. команды выбора шрифта или стиля шрифта: жирный, наклонный, курсив и т.п.;
3. команды выбора размера шрифта;
4. команды перехода на следующую строку или новую страницу: `\newpage` или `\\`;
5. Вертикальные и горизонтальные отступы: `\vskip`, `\hskip`,
6. Пробелы разной ширины: `\:`, `\;`, `\quad` и т.п. `\hspace`, `\vspace` и т.п.
7. команды смены выравнивания: по левому краю, по правому краю, по центру и т.п.

Эти команды нужно избегать при написании текстов. Вместо этого нужно использовать команды, которые помечают смысл участков текста. Например, если необходимо выделить курсивом какое-то слово в тексте, это означает, что вы хотите, чтобы читатель обратил на него внимание. И правильная команда будет команда `\emph{важное слово}`, которая говорит, что это место необходимо выделить. Чаще всего выделение будет сделано курсивом, но это не обязательно. Внутри курсива для выделения будет использоваться прямой шрифт, а если вы подключите пакет для презентаций `beamer`, выделение будет сделано цветом.

При написании конспекта необходимо использовать класс документа `lecture-notes`. В этом классе описано оформление конспектов. Задача автора конспекта набрать текст и правильно выделить в нем смысловые участки: разделы, подразделы, теоремы, доказательства, замечания, определения.

Описание внешнего вида этих участков — шрифтов, отступов, требований к положению на листе, находится внутри класса.

Такое выделение оформления в класс позволит сделать все конспекты единообразными, и позволит одновременно изменять оформление всех конспектов. Такой же стиль написания текстов применим и при использовании офисных продуктов типа Word. В Word можно задавать абзацам стили и настраивать отдельно каждый стиль абзаца: шрифт, отступы, выравнивание. Я рекомендую научиться использовать Word именно таким образом вместо того, чтобы давать явные команды на изменение шрифтов. Это избавляет от очень распространенных ошибок в оформлении Word документов, когда разные части документа набраны разными шрифтами. Допустим, в одном месте картинка подписана в Times New Roman, а в другом — Arial. Имейте привычку задавать абзацу стиль вместо того, чтобы вручную его форматировать.

#### **6.1.1 Что если все-таки нужно настроить оформление**

Если в конспекте требуется какая-то команда, управляющая оформлением, например, добавляющая вертикальный отступ, остановитесь и подумайте. Действительно, отделить оформление возможно не всегда, выше описана идеальная ситуация, когда оформление находится внутри класса, а в конспекте находится только содержание. Совершенно нормально, что  $\text{\TeX}$  не смог где-то хорошо отобразить текст, и требуется ему помочь, например, правильным отступом.

Возможно, проблема решается правильной расстановкой смысловых блоков. Например, отступа не хватает, потому что вы забыли написать команду начала раздела или не указали, что вы пишете определение.

Другая ситуация, если вы поймете, что появился новый, не учтенный в классе смысловой блок. Допустим, в отдельной лекции лектор часто вставляет рекламу других курсов. В этом случае необходимо самостоятельно завести блок «реклама», аналогичный блоку «замечание» или «теорема». Это делается один раз в заголовке документа. Возможно, со временем этот блок добавится в класс, если окажется, что такой блок нужен для многих конспектов.

Последняя ситуация — это когда действительно нужна единократная команда смены шрифта или установки отступа. И больше подобные ситуации в конспекте не ожидаются. В этом случае вставляйте случайную команду оформления прямо в текст, к сожалению, это распространенная ситуация. Только следите, что будет происходить дальше. Если вы поймете, что в конспекте нужно много раз сделать аналогичное исправление, значит, все же есть какой-то смысл за этим исправлением, и необходимо ввести для него собственную команду или собственное окружение.

## 6.2 Математический режим

В тексте важно вовремя включать математический режим. Перейти в режим можно либо с помощью долларов  $\$формула\$$  для формул внутри строки, либо командами  $\backslash[формула\]$  для выключенных формул, т.е. формул на отдельной строке. Обратите внимание, что при написании конспекта нужно использовать именно такой способ оформления формул, потому что стандартный способ через двойные доллары  $\$формула\$\$$  недостаточно гибок. Команду  $\backslash[$  можно переопределять и настраивать, например, можно заставить ее нумеровать формулы.

Математический режим помимо прочего переходит на отдельный математический шрифт. Поэтому важно переходить в этот режим при вводе всех математических объектов. Например, лучше писать

Решите уравнение и найдите  $x$

а не

Решите уравнение и найдите  $x$

потому что во втором случае  $x$  будет набран неправильным шрифтом.

## 6.3 Пробелы в математическом режиме

В математическом режиме все введенные пользователем пробелы игнорируются, и правильные пробелы расставляются автоматически. Поэтому важно правильно выбирать математические команды, чтобы L<sup>A</sup>T<sub>E</sub>X понял смысл формулы. Несколько примеров для пояснения.

Неправильно. Здесь <i>sin</i> это как будто произведение трех переменных	<code>\$f(x) = sin(x)\$</code>	$f(x) = sin(x)$
Правильно. Здесь <code>\sin</code> это функция, она набирается прямым шрифтом	<code>\$f(x) = \sin(x)\$</code>	$f(x) = \sin(x)$
Неправильно. Для операции деления с остатком выбран неправильный оператор <code>\mod</code> из пакета <code>amsmath</code> , обратите внимание на длинный пробел	<code>\$10 \mod 3 = 1\$</code>	$10 \mod 3 = 1$
Правильно. Для операции деления с остатком выбран оператор <code>\bmod</code> , который является бинарным оператором	<code>\$10 \bmod 3 = 1\$</code>	$10 \bmod 3 = 1$
Кстати, есть еще оператор <code>\pmod</code>	<code>\$10 \equiv 1 \pmod 3\$</code>	$10 \equiv 1 \pmod 3$

## 6.4 Математические символы

Необходимо правильно выбирать и математические символы. Очень часто нужны и часто вводятся неправильно следующие символы:

Явное умножение	<code>\$x\cdot y\$</code> или <code>\$A\times B\$</code>	$x \cdot y$ или $A \times B$
Многоточие	<code>\$x_1, x_2, \ldots, x_n\$</code>	$x_1, x_2, \dots, x_n$
Многоточие по середине строки	<code>\$A_1 \times A_2 \times \cdots \times A_n\$</code>	$A_1 \times A_2 \times \cdots \times A_n$
Следствие внутри доказательства	<code>\$x &gt; 1 \implies x &gt; 0\$</code>	$x > 1 \implies x > 0$
Равносильность внутри доказательства	<code>\$x &gt; 1 \iff x - 1 &gt; 0\$</code>	$x > 1 \iff x - 1 > 0$

Лучше всего, если вы просмотрите один раз, какие вообще бывают математические символы. Полный список символов по ссылке <http://tug.ctan.org/info/symbols/comprehensive/symbols-a4.pdf>. Когда посмотритесь, используйте более короткие и практические списки символов по ссылкам: <http://webhome.auburn.edu/~tamtiny/Symbols.pdf> или [https://oeis.org/wiki/List\\_of\\_LaTeX\\_mathematical\\_symbols](https://oeis.org/wiki/List_of_LaTeX_mathematical_symbols).

## 6.5 Кавычки в русском

Кавычки в русском языке бывают двух видов: «елочки» и „лапки”. Они набираются как `<<елочки>>` и `,,лапки,,`. Не используйте символ `"`, поэто-

му что он не подходит для текстов на русском. Лапки нужно использовать только как кавычки внутри кавычек, в остальных случаях используйте елочки. Почитайте в википедии про кавычки в русском языке, чтобы узнать об этом подробнее.

Если вы можете ввести кавычки елочки напрямую с клавиатуры вместо двух символов <<, используйте это. Это же относится и к другим символам, например, если можете сразу написать ä, пишите, если нет, вводите \"a.

## 6.6 Дефис, тире, минус, диапазон

Символы дефис, тире, диапазон, минус — это четыре разных символа. Они набираются как -, --, ---, - (в математическом режиме) соответственно. Ну или они вводятся как отдельный символ, если на вашей клавиатуре можно такой символ набрать. Сравните: **сине-зелёный**, **страницы 100--200**, **конь~---** это не **мышь**, **\$2-3=-1\$**.

## 6.7 Неразрывный пробел

Символ ~ (тильда) используется как неразрывный пробел. Его надо ставить перед тире, чтобы тире не оттянулось далеко от предыдущего слова и не перенеслось на следующую строку:

Число~--- это когда чего-то несколько.

Аналогично, его надо ставить перед указанием номера рисунка, ссылкой и т.п.: рисунок~\ref{fig:1}.

## 6.8 Теоремы, замечания, доказательства

В конспектах часто встречаются структурные блоки типа теорем, лемм, доказательств, замечаний, задач. Многие такие блоки уже настроены в классе документа, например, можно писать

```
\begin{theorem}[Основная теорема арифметики]
Условие теоремы
\end{theorem}
\begin{proof}
Доказательство
\end{proof}
```

Все возможные окружения описаны в шаблоне конспекта. Там же указаны подробности о том, как включать или отключать нумерацию блоков. Если какого-то типа блока не хватает, вы должны ввести его сами, это несложно и описано по ссылке [https://www.overleaf.com/learn/latex/theorems\\_and\\_proofs](https://www.overleaf.com/learn/latex/theorems_and_proofs).



## 6.9 Алгоритмы

Перед написанием алгоритма посмотрите, не лучше ли будет ввести его с помощью одного из стандартных пакетов <https://www.overleaf.com/learn/latex/algorithms>. По ссылке описаны пакеты `algorithmic` и `algorithm2e` для ввода псевдокода, выбор одно из них — дело вкуса. И пакет `listings` для вывода листингов кода.