

Презентация о презентации

Маленький такой подзаголовок

Никита Колганов

Московский физико-технический институт
nikita.kolganov@phystech.edu

Уже 15 ноября 2020 г., пора начинать ботать

Как устроен этот файл

DISCLAIMER: Если вам не интересно, как и почему всё работает, пропустите этот слайд.

Как и каждый латеховский проект, этот документ начинается с шапки

%класс документа - beamer, специальный класс для презентаций

```
\documentclass{beamer}
```

%далее идут импорты необходимых пакетов

```
\usepackage[T2A]{fontenc}      %шрифт
```

```
\usepackage[utf8]{inputenc}    %кодировка
```

```
\usepackage[russian]{babel}    %обработка русского текста
```

```
\usepackage{graphicx}          %графика
```

```
\usepackage{minted}            %листинги кода, типа этого
```

Информация о докладчике и докладе

Далее идут команды, отображаемые на титульном слайде вашей презентации

```
\title[Короткое название]{Название презентации}  
\subtitle{Подзаголовок презентации}  
\author[Сокращенное имя докладчика]{Имя докладчика}  
\institute[Короткое название организации]  
           {Полное название организации}  
\date[Короткая дата]{Дата}      %можно поставить \today
```

Здесь и везде – выражения в квадратных скобках – необязательные, их можно не заполнять (как и сами скобки).

Настройка темы презентации

Следующая команда задаёт тему презентации. Они называются по городам. Я рекомендую Warsaw и Madrid (никогда там не был).

```
\usetheme{Madrid}
```

В презентациях принято использовать шрифты “без засечек”, они хорошо читаются издалека. Однако формулы, написанные шрифтом без засечек выглядят отвратительно. Исправить это недоразумение можно следующей командой.

```
\usefonttheme[onlymath]{serif}
```

Тело документа

Далее начинается “тело документа” — то, что будет отображаться

```
\begin{document}
```

...

```
\end{document}
```

Все, что написано вне этого окружения по умолчанию игнорируется.

Вообще, примерно так выглядят все окружения в \LaTeX

```
\begin{имя окружения}[необязательные аргументы]  
                        {обязательные (если предусмотрены)}
```

Какая-то начинка

```
\end{имя окружения}
```

С другими окружениями мы познакомимся на следующем слайде.

Чтобы создать слайд, нужно (внутри окружения `document`) написать

```
\begin{frame}{Название слайда}
```

Содержание слайда

```
\end{frame}
```

Слайд о первом слайде

Чтобы создать титульный слайд (с информацией о докладе и докладчике), нужно использовать команду `maketitle` внутри окружения слайда `frame`

```
\begin{frame}  
\titlepage  
\end{frame}
```

В этом случае указывать название слайда не нужно.

Чтобы разбить слайд на колонки, нужно использовать окружение `columns`

```
\begin{columns}
```

```
\column{0.6\textwidth}
```

 Колонка, шириной 0.6 от ширины слайда.
Этот текст является содержимым первой колонки.

```
\column{0.4\textwidth}
```

 Колонка, шириной 0.4 от ширины слайда
Этот текст является содержимым второй колонки.

```
\end{columns}
```

Результат --- на следующем слайде.

Колоночки — демонстрация

Текст до колоночек
Колонка, шириной 0.4 от
ширины слайда. Этот текст
является содержимым
первой колонки.

Текст после колоночек

Колонка, шириной 0.5 от ширины слайда
Этот текст является содержимым второй
колонки.

Картиночки

Для добавления картинок используется окружение `figure`. Но оно создаёт только поле для картинки/её названия, для добавления самой картинки нужно использовать `includegraphics`

Результат:

Код:

```
\begin{figure}
\centering %центрирование
\includegraphics[width=0.7\textwidth]
{abacaba.png}
\caption{Подпись к картинке}
\end{figure}
```

Мы задействовали необязательный аргумент `includegraphics`, настраивающий ширину картинки.

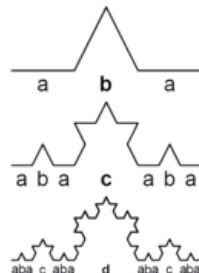


Рис.: Подпись к картинке

Листинги кода

А теперь один из самых важных моментов для презентации вашего проекта — листинги кода. Чтобы beamer не портил форматирование кода, необходимо использовать специальное окружение — `verbatim`. Оно воспроизводит содержащийся в нём текст “буквально”

```
\begin{verbatim}
```

Текст, написанный внутри этого окружения
воспроизводится буквально, с сохранением
пробелов
и переносов

```
\begin{verbatim}
```

Однако, чтобы использовать это окружение внутри слайда, надо обязательно использовать необязательный аргумент `fragile`!

```
\begin{frame}[fragile]{<-- необязательный аргумент тут}
```

Содержание слайда

```
\end{frame}
```

Листинги кода – upper intermediate

Есть специальный пакет — `minted`, который не только сохраняет форматирование кода, но и подсвечивает его, согласно синтаксису конкретного языка.

Так выглядит код:

```
\begin{minted}{python}
def funksiya():
    return "tak nazyvat' funktsii nel'zya"
\end{minted}
```

А так — результат:

```
def funksiya():
    return "tak nazyvat' funktsii nel'zya"
```

Вместо `python` можно указать другой язык, например `C++` или `latex`.

Если вы пользуетесь overleaf.com, то латеховский код, использующий окружение `minted` скомпилируется без проблем. Однако, чтобы всё заработало в `miktex`, необходимо лезть в настройки, а именно:

- 1 Нажать “Правка” — “Настройки”
- 2 Во вкладке “Вёрстка” в списке “Инструменты вёрстки” выбрать активный профиль вёрстки (скорее всего это `pdfLaTeX+MakeIndex+BibTeX` — можете посмотреть справа от зелёной стрелочки — кнопки, которой вы запускаете компиляцию вашего латеховского проекта)
- 3 Справа от списка нажать “Правка”. В появившемся окне нажать плюсики, после чего в списке “Параметры” нужно вписать “`--tex-option=--shell-escape`”. Выходите из окошек кнопкой “ОК”
- 4 Готово, можно пользоваться `minted`

Кстати, на предыдущем слайде появился список — для него тоже есть своё окружение `enumerate`. Очередной элемент списка начинается с команды `item`.

Код:

```
\begin{enumerate}  
\item Текст первого пункта  
\item Текст второго пункта  
\item Надоело  
\end{enumerate}
```

Результат:

- 1 Текст первого пункта
- 2 Текст второго пункта
- 3 Надоело

Для нумерованного списка вместо `enumerate` напишите `itemize`.

Формулки

Вообще, \LaTeX придумал великий математик Дональд Кнут, чтобы писать в нём красивые формулки. Формулы делятся на внутристрочные и вынесенные. Первые выглядят так $S = \pi r^2$, а вторые — так

$$G(x - x') = \int \frac{d^4 p}{(2\pi)^4} \frac{e^{ip_\mu(x-x')^\mu}}{p_\nu^2 - m^2 + i\epsilon}$$

Внутристрочные выделяются одиночными знаками доллара

`$S=\pi r^2$`, а вынесенные — двойными

`$$`

`G(x-x') = \int \frac{d^4 p}{(2 \pi)^4}`

`\frac{e^{i p_\mu (x - x')^\mu}}{p_\nu^2 - m^2 + i \epsilon}`

`$$`

Научиться писать красивые формулы не является целью этой презентации. Однако общий смысл и правила можно почерпнуть из примеров выше. Для большего есть прекрасные руководства, в том числе на русском языке.

Спасибо за понимание:)