

# Дополнительные пакеты

Учебная практика

мехмат, III курс, кафедра ИВЭ

**xcolor**

Драйверы 1

Драйверы 2

Драйверы 3

Драйверы 4

Драйверы 5

Принципы

Определение

Текст

Фон

Смещение

graphicx

hyperref

listings

datatool 1

datatool 2

# Пакет xcolor

# Поддержка цвета и графики в L<sup>A</sup>T<sub>E</sub>X

[xcolor](#)

**Драйверы 1**

[Драйверы 2](#)

[Драйверы 3](#)

[Драйверы 4](#)

[Драйверы 5](#)

[Принципы](#)

[Определение](#)

[Текст](#)

[Фон](#)

[Смещение](#)

[graphicx](#)

[hyperref](#)

[listings](#)

[datatool 1](#)

[datatool 2](#)

- ✓ Изначально система T<sub>E</sub>X разрабатывалась без учёта необходимости использования цвета и сложной графики в документах. Эти возможности предоставляются «драйверами печати», отвечающими за преобразование DVI-файла.
- ✓ Система T<sub>E</sub>X предоставляет возможность передачи драйверам информации, о которой ей самой ничего не известно, при помощи команд `\special`. Эта информация записывается непосредственно в DVI-файл. Аналогично работает система передачи информации механизму pdfT<sub>E</sub>X.

# Поддержка цвета и графики в $\LaTeX$ (продолжение)

xcolor

Драйверы 1

**Драйверы 2**

Драйверы 3

Драйверы 4

Драйверы 5

Принципы

Определение

Текст

Фон

Смещение

graphicx

hyperref

listings

datatool 1

datatool 2

- ✓ Далее эта информация интерпретируется драйвером. Например, включение внешнего рисунка приводит к записи в DVI-файл специфичных для драйвера команд вместе со ссылкой на файл с изображением. Драйвер должен включить этот файл в результирующий документ.
- ✓ Многие дополнительные возможности обрабатываются разными драйверами по-разному. Например, команды смены цвета фона могут воздействовать на следующие страницы или нет.

# Поддержка цвета и графики в $\LaTeX$ (продолжение)

xcolor

Драйверы 1

Драйверы 2

Драйверы 3

Драйверы 4

Драйверы 5

Принципы

Определение

Текст

Фон

Смещение

graphicx

hyperref

listings

datatool 1

datatool 2

- ✓ Авторы  $\LaTeX$  определили некоторые пакеты как «обязательный» набор: предполагается, что они должны быть частью любого дистрибутива  $\LaTeX$ . В его состав входят, например,  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\LaTeX$ , `babel`, а также пакеты `color` и `graphics`. Два последних пакета поддерживают возможность работы в  $\LaTeX$  с цветом и включением внешних рисунков независимым от используемого драйвера способом.
- ✓ Таким образом, для использования одного из этих пакетов ему необходимо передать параметр, сообщающий о намерении использовать тот или иной драйвер.

Пример

```
\usepackage [dvipdfm] {color}
```

# Поддержка цвета и графики в $\LaTeX$ (продолжение)

xcolor

Драйверы 1

Драйверы 2

Драйверы 3

Драйверы 4

Драйверы 5

Принципы

Определение

Текст

Фон

Смещение

graphicx

hyperref

listings

datatool 1

datatool 2

- ✓ Чаще всего требуется создавать документы, которые подготавливаются либо при помощи pdf $\TeX$ , либо dvips. В таком случае можно подготовить универсальный файл, подключаемый в преамбуле и автоматически определяющий режим компиляции документа при помощи пакета ifpdf.

# Поддержка цвета и графики в L<sup>A</sup>T<sub>E</sub>X (окончание)

xcolor

Драйверы 1

Драйверы 2

Драйверы 3

Драйверы 4

**Драйверы 5**

Принципы

Определение

Текст

Фон

Смешение

graphicx

hyperref

listings

datatool 1

datatool 2

```
% Подключить до \documentclass {...}
% \input {select-driver}

\RequirePackage {ifpdf}

\ifpdf
  % Ничего
\else
  \PassOptionsToPackage {dvips} {graphics}
  \PassOptionsToPackage {dvips} {hyperref}
  \PassOptionsToPackage {dvips} {xcolor}
\fi
```

# Принципы работы пакета xcolor

xcolor

Драйверы 1

Драйверы 2

Драйверы 3

Драйверы 4

Драйверы 5

**Принципы**

Определение

Текст

Фон

Смещение

graphicx

hyperref

listings

datatool 1

datatool 2

- ✓ Пакет загружает пакет color и определяет дополнительные команды работы с цветами и т. д.
- ✓ Предоставляет набор именованных цветов (**white**, **green** и т. д.) Новые имена можно определять при помощи одной из цветовых моделей (rgb, cmyk, hsb, HTML и т. д.) либо определять безымянные цвета непосредственно в месте их использования.
- ✓ Цвета можно использовать в командах выбора цвета текста и фона, определяемых пакетом, а также передавать другим пакетам, поддерживающим цвета.

# Определение цвета

xcolor

Драйверы 1

Драйверы 2

Драйверы 3

Драйверы 4

Драйверы 5

Принципы

**Определение**

Текст

Фон

Смещение

graphicx

hyperref

listings

datatool 1

datatool 2

**\definecolor** {⟨имя⟩} {⟨модель⟩} {⟨определение⟩}

Определяет новый цвет с заданным именем.

Пример

```
\definecolor {gray-red} {rgb} {1, 0.5, 0.5}  
\definecolor {myblue} {HTML} {6060E7}
```

*Замечание:* все команды пакета, использующие цвета, имеют две формы:

```
\⟨команда⟩ {⟨имя_цвета⟩} ...  
\⟨команда⟩ [⟨модель⟩] {⟨определение_цвета⟩} ...
```

Первый вариант использует имя ранее определённого цвета, второй — непосредственное определение. △

# Установка цвета текста и страницы

[xcolor](#)

[Драйверы 1](#)

[Драйверы 2](#)

[Драйверы 3](#)

[Драйверы 4](#)

[Драйверы 5](#)

[Принципы](#)

[Определение](#)

**Текст**

[Фон](#)

[Смещение](#)

[graphicx](#)

[hyperref](#)

[listings](#)

[datatool 1](#)

[datatool 2](#)

**\color** [*⟨модель⟩*] {⟨имя\_или\_определение⟩}

Устанавливает цвет текста в текущей группе, который выбирается после команды `\selectfont`.

**\textcolor** [*⟨модель⟩*] {⟨цвет\_текста⟩} {⟨текст⟩}

Печатает заданный текст заданным цветом.

Пример

```
\textcolor {orange} {Test}
```

Test

**\pagecolor** [*⟨модель⟩*] {⟨цвет\_страницы⟩}

Устанавливает цвет текущей и всех последующих страниц.

# Установка цвета фона и рамки

xcolor

Драйверы 1

Драйверы 2

Драйверы 3

Драйверы 4

Драйверы 5

Принципы

Определение

Текст

**Фон**

Смещение

graphicx

hyperref

listings

datatool 1

datatool 2

**\colorbox** [*⟨модель⟩*] {⟨имя\_или\_определение⟩}

Печатает текст в прямоугольнике с заданным фоном.

Пример

```
\colorbox {orange} {Test}
```



**\fcolorbox** [*⟨модель<sub>1</sub>⟩*] {⟨цвет\_рамки⟩} [*⟨модель<sub>2</sub>⟩*]  
{⟨цвет\_фона⟩} {⟨текст⟩}

Печатает текст в прямоугольнике с рамкой заданного цвета с заданным фоном.

Пример

```
\fcolorbox {purple} {orange} {Test}
```



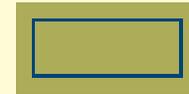
# Смешение цветов

*Замечание:* Пакет предоставляет возможность определять цвета при помощи смешивания уже существующих цветов в заданной пропорции.



## Пример

```
\colorbox {red!50!green!65!} %  
{\boxframe {1cm} {1em} {0em}}
```



Здесь смешиваются красный и зелёный цвета в пропорции 50 % на 50 %, результат смешивается с белым цветом (по умолчанию) в пропорции 65 % на 35 %.

- xcolor
- Драйверы 1
- Драйверы 2
- Драйверы 3
- Драйверы 4
- Драйверы 5
- Принципы
- Определение
- Текст
- Фон
- Смешение**
- graphicx
- hyperref
- listings
- datatool 1
- datatool 2

xcolor

**graphicx**

Принципы 1

Принципы 2

Принципы 3

PNG/EPS

PNG/EPS 2

Пример

Подключение

Параметры 1

Параметры 2

hyperref

listings

datatool 1

datatool 2

# Пакет `graphicx`

# Принципы работы пакета `graphicx`

[xcolor](#)

[graphicx](#)

**Принципы 1**

Принципы 2

Принципы 3

PNG/EPS

PNG/EPS 2

Пример

Подключение

Параметры 1

Параметры 2

[hyperref](#)

[listings](#)

[datatool 1](#)

[datatool 2](#)

- ✓ Пакет предоставляет удобный интерфейс к пакету `graphics` (подключение внешних графических файлов) в виде пар ключей и значений.
- ✓ Драйвер `pdfTEX` поддерживает включение растровых изображений (JPEG, PNG и т. д.), драйвер `dvips` — векторных в формате EPS. Изображения формата EPS можно преобразовать к PDF для возможности использования их драйвером `pdfTEX`.

# Принципы работы пакета `graphicx` (продолжение)

- [xcolor](#)
- [graphicx](#)
- [Принципы 1](#)
- [Принципы 2](#)**
- [Принципы 3](#)
- [PNG/EPS](#)
- [PNG/EPS 2](#)
- [Пример](#)
- [Подключение](#)
- [Параметры 1](#)
- [Параметры 2](#)
- [hyperref](#)
- [listings](#)
- [datatool 1](#)
- [datatool 2](#)

- ✓ Чтобы драйвер `dvips` мог подключать растровые изображения, можно при помощи утилиты `bmeeps` сгенерировать `bb`-файлы на языке EPS с информацией об их размерах. Пример содержимого файла `picture.bb`, построенное по файлу `picture.png` при помощи команды:

```
bmeeps -b picture.png picture.bb
```

Пример (`picture.bb`)

```
%%BoundingBox: 0 0 292 113
```

# Принципы работы пакета `graphicx` (окончание)

[xcolor](#)

[graphicx](#)

[Принципы 1](#)

[Принципы 2](#)

[Принципы 3](#)

[PNG/EPS](#)

[PNG/EPS 2](#)

[Пример](#)

[Подключение](#)

[Параметры 1](#)

[Параметры 2](#)

[hyperref](#)

[listings](#)

[datatool 1](#)

[datatool 2](#)

- ✓ Далее необходимо в преамбуле выполнить следующие команды:

Пример

```
\ifpdf
  % Nothing
\else
  \DeclareGraphicsRule {.png} {eps} {.bb} {}
\fi
```

- ✓ В этом случае для подключения растровых изображений в формате PNG будут работать одни и те же команды для драйверов pdfTeX и dvips.

# Одновременная поддержка драйверов dvips и pdftex

xcolor

graphicx

Принципы 1

Принципы 2

Принципы 3

**PNG/EPS**

PNG/EPS 2

Пример

Подключение

Параметры 1

Параметры 2

hyperref

listings

datatool 1

datatool 2

*Замечание:* Современные дистрибутивы  $\text{T}_{\text{E}}\text{X}$  могут не поддерживать подключение растровых изображений с драйвером dvips. В этом случае можно преобразовать изображение из формата PNG в формат EPS, например, при помощи графического редактора GIMP.



# Одновременная поддержка драйверов dvips и pdftex (окончание)

[xcolor](#)

[graphics](#)

Принципы 1

Принципы 2

Принципы 3

PNG/EPS

**PNG/EPS 2**

Пример

Подключение

Параметры 1

Параметры 2

[hyperref](#)

[listings](#)

[datatool 1](#)

[datatool 2](#)

*Замечание:* Чтобы обеспечить совместимость документа одновременно с драйверами dvips и pdftex, можно хранить обе версии графических файлов и при их подключении не указывать расширения. Для добавления поддерживаемых пакетом graphics расширений по умолчанию можно воспользоваться командой:

```
\DeclareGraphicsExtensions {⟨список_расширений⟩}
```



# Пример одновременной поддержки драйверов

[xcolor](#)

[graphicx](#)

[Принципы 1](#)

[Принципы 2](#)

[Принципы 3](#)

[PNG/EPS](#)

[PNG/EPS 2](#)

**Пример**

[Подключение](#)

[Параметры 1](#)

[Параметры 2](#)

[hyperref](#)

[listings](#)

[datatool 1](#)

[datatool 2](#)

```
\documentclass {article}

\usepackage {ifpdf}
\ifpdf
  % Ничего
\else
  \PassOptionsToPackage {dvips} {graphics}
\fi

\usepackage {graphicx}
\ifpdf
  \DeclareGraphicsExtensions {.pdf, .png, .jpg}
\fi

\begin {document}
  \includegraphics [width = \textwidth] {picture}
\end {document}
```

# Подключение файлов

[xcolor](#)

[graphicx](#)

Принципы 1

Принципы 2

Принципы 3

PNG/EPS

PNG/EPS 2

Пример

**Подключение**

Параметры 1

Параметры 2

[hyperref](#)

[listings](#)

[datatool 1](#)

[datatool 2](#)

**`\graphicspath`** { $\langle$ список\_каталогов $\rangle$ }

Определяет список путей к каталогам, каждый путь в фигурных скобках, относительно которых команда `\includegraphics` будет искать заданные файлы.

Пример

```
\graphicspath {{images}{screenshots}}
```

**`\includegraphics`** [ $\langle$ параметры $\rangle$ ] { $\langle$ файл $\rangle$ }

Подключает изображение из заданного файла.

# Основные параметры команды `\includegraphics`

[xcolor](#)

[graphicx](#)

[Принципы 1](#)

[Принципы 2](#)

[Принципы 3](#)

[PNG/EPS](#)

[PNG/EPS 2](#)

[Пример](#)

[Подключение](#)

**[Параметры 1](#)**

[Параметры 2](#)

[hyperref](#)

[listings](#)

[datatool 1](#)

[datatool 2](#)

**width** =  $\langle \text{ширина} \rangle$

Определяет ширину изображения в тексте.

**height** =  $\langle \text{высота} \rangle$

Определяет высоту изображения в тексте. Хотя бы один из параметров `width`, `height` нужно обязательно указать.

**keepaspectratio** = `true`, `false` (по умолчанию: **true**)

Определяет, нужно ли сохранять исходно соотношение сторон изображения в документе.

**draft** = `true`, `false` (по умолчанию: **false**)

Также может быть указан глобально в настройках пакета. Не включает изображение, оставляет прямоугольную рамку нужного размера с именем файла.

# Основные параметры команды `\includegraphics` (окончание)

[xcolor](#)

[graphicx](#)

Принципы 1

Принципы 2

Принципы 3

PNG/EPS

PNG/EPS 2

Пример

Подключение

Параметры 1

**Параметры 2**

[hyperref](#)

[listings](#)

[datatool 1](#)

[datatool 2](#)

**angle** =  $\langle \text{угол} \rangle$

Угол поворота.

**origin** =  $\langle \text{координаты} \rangle$

Координаты центра поворота.

Пример

```
\includegraphics %  
[ %  
  width = 0.7 \textwidth %  
] %  
{images/01.png}
```

xcolor

graphicx

**hyperref**

Принципы

Цвет 1

Цвет 2

PDF 1

PDF 2

Отображение 1

Отображение 2

Свойства

Пример

Ссылки 1

Ссылки 2

Ссылки 3

Ссылки 4

Закладки

listings

datatool 1

datatool 2

# Пакет `hyperref`

# Принципы работы пакета `hyperref`

[xcolor](#)

[graphicx](#)

[hyperref](#)

**Принципы**

[Цвет 1](#)

[Цвет 2](#)

[PDF 1](#)

[PDF 2](#)

[Отображение 1](#)

[Отображение 2](#)

[Свойства](#)

[Пример](#)

[Ссылки 1](#)

[Ссылки 2](#)

[Ссылки 3](#)

[Ссылки 4](#)

[Закладки](#)

[listings](#)

[datatool 1](#)

[datatool 2](#)

- ✓ Пакет предоставляет команды для добавления в документ гиперссылок. Также он переопределяет стандартные команды  $\LaTeX$  перекрёстных ссылок (`\tableofcontents`, `\cite`, `\ref` и т. д.), а также команды пакета `url` и т. д., дополняя их командами `\special`, которые драйвер способен превратить в гиперссылки в результирующем документе.
- ✓ Также позволяет управлять внешним видом результирующего документа, поддерживая возможности формата PDF: генерирование PDF-закладок на основе оглавления документа  $\LaTeX$ , управление начальным внешним видом страницы в программе просмотра, определение диалоговых форм и т. д.

# Основные параметры цвета

[xcolor](#)

[graphicx](#)

[hyperref](#)

Принципы

**Цвет 1**

Цвет 2

PDF 1

PDF 2

Отображение 1

Отображение 2

Свойства

Пример

Ссылки 1

Ссылки 2

Ссылки 3

Ссылки 4

Закладки

[listings](#)

[datatool 1](#)

[datatool 2](#)

**`\hypersetup`** {*⟨параметры⟩*}

Параметры можно задавать этой командой или передавать их в команду `\usepackage`. Основные поддерживаемые параметры:

**`colorlinks`** = `true`, `false` (по умолчанию: **`false`**)

Отмечать в документе текст гиперссылок цветом.

# Основные параметры цвета (окончание)

- [xcolor](#)
- [graphicx](#)
- [hyperref](#)
- Принципы
- Цвет 1
- Цвет 2**
- PDF 1
- PDF 2
- Отображение 1
- Отображение 2
- Свойства
- Пример
- Ссылки 1
- Ссылки 2
- Ссылки 3
- Ссылки 4
- Закладки
- [listings](#)
- [datatool 1](#)
- [datatool 2](#)

**linkcolor** =  $\langle \text{цвет} \rangle$  (по умолчанию: **red**)

Цвет обычных ссылок.

**anchorcolor** =  $\langle \text{цвет} \rangle$  (по умолчанию: **black**)

Цвет текста, являющегося целью гиперссылок.

**citecolor** =  $\langle \text{цвет} \rangle$  (по умолчанию: **green**)

Цвет библиографических цитат.

**filecolor** =  $\langle \text{цвет} \rangle$  (по умолчанию: **cyan**)

Цвет URL на локальные файлы.

**urlcolor** =  $\langle \text{цвет} \rangle$  (по умолчанию: **magenta**)

Цвет URL.

# Основные параметры, специфичные для PDF

[xcolor](#)

[graphicx](#)

[hyperref](#)

Принципы

Цвет 1

Цвет 2

**PDF 1**

PDF 2

Отображение 1

Отображение 2

Свойства

Пример

Ссылки 1

Ссылки 2

Ссылки 3

Ссылки 4

Закладки

[listings](#)

[datatool 1](#)

[datatool 2](#)

**bookmarks** = true, false (по умолчанию: **true**)

Генерировать закладки PDF на основе оглавления.

**bookmarksopen** = true, false (по умолчанию: **false**)

Показывать дерево закладок в развёрнутом виде при открытии документа.

**bookmarksopenlevel** = *номер*

Максимальный уровень вложенности, до которого будут открыты закладки.

# Основные параметры, специфичные для PDF (окончание)

[xcolor](#)

[graphicx](#)

[hyperref](#)

Принципы

Цвет 1

Цвет 2

PDF 1

**PDF 2**

Отображение 1

Отображение 2

Свойства

Пример

Ссылки 1

Ссылки 2

Ссылки 3

Ссылки 4

Закладки

[listings](#)

[datatool 1](#)

[datatool 2](#)

**bookmarknumbered** = true, false (по умолчанию: **false**)

Включать номера разделов в текст закладок.

**unicode** = true, false (по умолчанию: **false**)

Хранить строки PDF в кодировке Unicode. Необходимо для корректного отображения закладок.

# Основные параметры отображения документа PDF

- [xcolor](#)
- [graphicx](#)
- [hyperref](#)
- Принципы
- Цвет 1
- Цвет 2
- PDF 1
- PDF 2
- Отображение 1**
- Отображение 2
- Свойства
- Пример
- Ссылки 1
- Ссылки 2
- Ссылки 3
- Ссылки 4
- Закладки
- [listings](#)
- [datatool 1](#)
- [datatool 2](#)

**pdfpragemode** =  $\langle \text{текст} \rangle$  (по умолчанию: пусто)

Определяет внешний вид документа при открытии.

Таблица 1: основные значения ключа pdfpragemode

Имя	Значение
пусто или <b>UseOutlines</b>	Показывать панель закладок.
<b>UseNone</b>	Не отображать дополнительных панелей.
<b>UseThumbs</b>	Показывать панель макетов страниц.
<b>FullScreen</b>	Использовать полноэкранный режим.

# Основные параметры отображения документа PDF (окончание)

[xcolor](#)

[graphicx](#)

[hyperref](#)

Принципы

Цвет 1

Цвет 2

PDF 1

PDF 2

Отображение 1

**Отображение 2**

Свойства

Пример

Ссылки 1

Ссылки 2

Ссылки 3

Ссылки 4

Закладки

[listings](#)

[datatool 1](#)

[datatool 2](#)

**pdfstartview** =  $\langle \text{текст} \rangle$  (по умолчанию: **Fit**)

Определяет внешний вид страницы при открытии документа.

Таблица 2: основные значения ключа pdfstartview

Имя	Значение
<b>Fit</b>	Подгонять размер страницы для её отображения полностью в окне.
<b>FitH</b>	Подгонять ширину страницы под окно.
<b>FitV</b>	Подгонять высоту страницы под окно.

# Основные параметры свойств документа PDF

[xcolor](#)

[graphicx](#)

[hyperref](#)

Принципы

Цвет 1

Цвет 2

PDF 1

PDF 2

Отображение 1

Отображение 2

**Свойства**

Пример

Ссылки 1

Ссылки 2

Ссылки 3

Ссылки 4

Закладки

[listings](#)

[datatool 1](#)

[datatool 2](#)

**pdftitle** =  $\langle \text{текст} \rangle$  (по умолчанию: пусто)

Название документа, отображаемое в его свойствах.

**pdfauthor** =  $\langle \text{текст} \rangle$  (по умолчанию: пусто)

Имена авторов документа, отображаемые в его свойствах.

**pdfsubject** =  $\langle \text{текст} \rangle$  (по умолчанию: пусто)

Тематика документа, отображаемая в его свойствах.

**pdfkeywords** =  $\langle \text{текст} \rangle$  (по умолчанию: пусто)

Набор ключевых слов документа, отображаемый в его свойствах.

# Пример использования пакета hyperref

xcolor

graphicx

hyperref

Принципы

Цвет 1

Цвет 2

PDF 1

PDF 2

Отображение 1

Отображение 2

Свойства

**Пример**

Ссылки 1

Ссылки 2

Ссылки 3

Ссылки 4

Закладки

listings

datatool 1

datatool 2

```
\usepackage [unicode] {hyperref}
```

```
\hypersetup %
```

```
{ %
```

```
  pdfpagemode = UseOutlines, %
```

```
  pdfstartview = FitH, %
```

```
  colorlinks, %
```

```
  pdfauthor = {Иванов И. И.}, %
```

```
} %
```

# Команды гиперссылок

[xcolor](#)

[graphicx](#)

[hyperref](#)

Принципы

Цвет 1

Цвет 2

PDF 1

PDF 2

Отображение 1

Отображение 2

Свойства

Пример

**Ссылки 1**

Ссылки 2

Ссылки 3

Ссылки 4

Закладки

[listings](#)

[datatool 1](#)

[datatool 2](#)

**`\hypertarget`** { $\langle$ имя\_ссылки $\rangle$ } { $\langle$ текст $\rangle$ }

Выводит заданный  $\langle$ текст $\rangle$  (произвольный код  $\LaTeX$ ) и делает его целью внутридокументной гиперссылки с заданным именем ( $\langle$ имя\_ссылки $\rangle$ ).

**`\hyperlink`** { $\langle$ имя\_ссылки $\rangle$ } { $\langle$ текст $\rangle$ }

Превращает заданный  $\langle$ текст $\rangle$  во внутридокументную гиперссылку на цель с заданным именем ( $\langle$ имя\_ссылки $\rangle$ ).

## Пример

```
\hyperlink {hyper:text} {Ссылка}
```

```
\clearpage
```

```
\hypertarget {hyper:text} {Текст}
```

# Команды гиперссылок (продолжение)

- [xcolor](#)
- [graphicx](#)
- [hyperref](#)
- Принципы
- Цвет 1
- Цвет 2
- PDF 1
- PDF 2
- Отображение 1
- Отображение 2
- Свойства
- Пример
- Ссылки 1
- Ссылки 2**
- Ссылки 3
- Ссылки 4
- Закладки
- [listings](#)
- [datatool 1](#)
- [datatool 2](#)

## `\url` { $\langle URL \rangle$ }

Выводит заданный  $\langle URL \rangle$  в виде глобальной гиперссылки, определяемой тем же самым  $\langle URL \rangle$ . Шрифт гиперссылок определяется символом `\UrlFont`, который определяет команды выбора шрифта.

### Пример

```
\def \UrlFont {\tt}
See \url {http://google.com/}
```

See <http://google.com/>

## `\nolinkurl` { $\langle URL \rangle$ }

Аналогично `\url`, однако не превращает выводимый текст в гиперссылку.

# Команды гиперссылок (продолжение)

[xcolor](#)

[graphicx](#)

[hyperref](#)

Принципы

Цвет 1

Цвет 2

PDF 1

PDF 2

Отображение 1

Отображение 2

Свойства

Пример

Ссылки 1

Ссылки 2

**Ссылки 3**

Ссылки 4

Закладки

[listings](#)

[datatool 1](#)

[datatool 2](#)

**`\hyperref`** [`<метка>`] {`<текст>`}

Выводит `<текст>`, превращая его в гиперссылку, указывающую туда же, куда указывает `\ref {<метка>}`.

**`\ref*`** {`<метка>`}

**`\pageref*`** {`<метка>`}

Аналогично `\ref` и `\pageref`, но не превращают выводимые метки в гиперссылки. Могут быть полезными в сочетании с командой `\hyperref`.

## Пример

Как было отмечено в `\hyperref [ch:intro] {главе~\ref* {ch:intro}}`,

# Команды гиперссылок (окончание)

[xcolor](#)

[graphicx](#)

[hyperref](#)

Принципы

Цвет 1

Цвет 2

PDF 1

PDF 2

Отображение 1

Отображение 2

Свойства

Пример

Ссылки 1

Ссылки 2

Ссылки 3

**Ссылки 4**

Закладки

[listings](#)

[datatool 1](#)

[datatool 2](#)

## **\phantomsection**

Добавляет в текущую позицию цель гиперссылки с пустым текстом. Бывает нужна для добавления закладок вручную при помощи команды `\pdfbookmark` и т. п.

## **\href** [*⟨настройки⟩*] {⟨URL⟩} {⟨текст⟩}

Выводит заданный *⟨текст⟩* и делает его целью глобальной гиперссылки, определяемой (*⟨URL⟩*). Основные настройки:

## **page** = *⟨номер⟩*

Определяет номер страницы во внешнем PDF-документе, начиная с 1.

# Команды определения PDF-закладок

[xcolor](#)

[graphicx](#)

[hyperref](#)

Принципы

Цвет 1

Цвет 2

PDF 1

PDF 2

Отображение 1

Отображение 2

Свойства

Пример

Ссылки 1

Ссылки 2

Ссылки 3

Ссылки 4

**Закладки**

[listings](#)

[datatool 1](#)

[datatool 2](#)

**\pdfbookmark** [*⟨уровень⟩*] {⟨текст⟩} {⟨имя⟩}

Создаёт закладку на текущую позицию в документе с заданным тестом на заданном уровне (по умолчанию 0), связывая с ней внутреннюю гиперссылку с заданным именем.

## Пример

```
\begin {document}

\pdfbookmark {Титульный лист} {bm:title}
\maketitle
\pdfbookmark {Оглавление} {bm:contents}
\tableofcontents
```

[xcolor](#)

[graphicx](#)

[hyperref](#)

**[listings](#)**

Назначение

Вывод 1

Вывод 2

Вывод 3

Настройки

Окружение

Пример 1

Стили

Диапазон

Язык

Текст

Фон

Символы

Пример 2

[datatool 1](#)

[datatool 2](#)

# Пакет listings

# Назначение пакета listings

[xcolor](#)

[graphicx](#)

[hyperref](#)

[listings](#)

**Назначение**

[Вывод 1](#)

[Вывод 2](#)

[Вывод 3](#)

[Настройки](#)

[Окружение](#)

[Пример 1](#)

[Стили](#)

[Диапазон](#)

[Язык](#)

[Текст](#)

[Фон](#)

[Символы](#)

[Пример 2](#)

[datatool 1](#)

[datatool 2](#)

- ✓ Пакет предназначен для набора текстов на различных языках программирования с возможностью автоматической подсветки синтаксиса, нумерации строк и т. д.
- ✓ Пакет представляет замену стандартного окружения `verbatim`, позволяющего набирать текст «как есть» без его интерпретации системой  $\text{T}_{\text{E}}\text{X}$ . Дополнительно поддерживаются другие возможности, например, вставка кода  $\text{T}_{\text{E}}\text{X}$  внутри обычного текста.

# Команды вывода

[xcolor](#)

[graphicx](#)

[hyperref](#)

[listings](#)

Назначение

**Вывод 1**

Вывод 2

Вывод 3

Настройки

Окружение

Пример 1

Стили

Диапазон

Язык

Текст

Фон

Символы

Пример 2

[datatool 1](#)

[datatool 2](#)

`\lstinline[параметры](символ)(текст)(символ)`

Печатает в тексте небольшой фрагмент кода.

Пример

The `\lstinline [language = C++]!virtual! keyword.`

The `virtual` keyword.

# Команды вывода (продолжение)

- [xcolor](#)
- [graphicx](#)
- [hyperref](#)
- [listings](#)
- Назначение
- Вывод 1
- Вывод 2**
- Вывод 3
- Настройки
- Окружение
- Пример 1
- Стили
- Диапазон
- Язык
- Текст
- Фон
- Символы
- Пример 2
- [datatool 1](#)
- [datatool 2](#)

**`\lstinputlisting`** [*⟨параметры⟩*] {*⟨файл⟩*}

Выводит содержимое указанного файла.

## Пример

```
\lstinputlisting %  
[ %  
  language = C++, %  
  linerange = {10 - 22} %  
] %  
  {listings/01.cpp}
```

# Команды вывода (окончание)

- [xcolor](#)
- [graphicx](#)
- [hyperref](#)
- [listings](#)
- Назначение
- Вывод 1
- Вывод 2
- Вывод 3**
- Настройки
- Окружение
- Пример 1
- Стили
- Диапазон
- Язык
- Текст
- Фон
- Символы
- Пример 2
- [datatool 1](#)
- [datatool 2](#)

```
\begin{lstlisting}[⟨параметры⟩]  
  ⟨содержимое_окружения⟩  
\end{lstlisting}
```

Печатает в тексте фрагмент кода, указанный внутри окружения.

## Пример

```
\begin{lstlisting}%  
  [language = C++, gobble = 2]  
  int main()  
  {  
    return 0;  
  }  
\end{lstlisting}
```

```
int main()  
{  
    return 0;  
}
```

# Команда определения глобальных настроек

[xcolor](#)

[graphicx](#)

[hyperref](#)

[listings](#)

Назначение

Вывод 1

Вывод 2

Вывод 3

**Настройки**

Окружение

Пример 1

Стили

Диапазон

Язык

Текст

Фон

Символы

Пример 2

[datatool 1](#)

[datatool 2](#)

**\lstset** {⟨параметры⟩}

Устанавливает настройки, глобально действующие на все распечатки далее в текущей группе.

## Пример

```
\lstset %  
  {language = C++, gobble = 2}  
  
\begin{lstlisting}  
  int main()  
  {  
    return 0;  
  }  
\end{lstlisting}
```

```
int main()  
{  
  return 0;  
}
```

# Команда определения окружения

- [xcolor](#)
- [graphicx](#)
- [hyperref](#)
- [listings](#)
- Назначение
- Вывод 1
- Вывод 2
- Вывод 3
- Настройки
- Окружение**
- Пример 1
- Стили
- Диапазон
- Язык
- Текст
- Фон
- Символы
- Пример 2
- [datatool 1](#)
- [datatool 2](#)

```
\lstnewenvironment {⟨имя_окружения⟩}  
[⟨количество_параметров⟩] [⟨значение_по_умолчанию⟩]  
{⟨начальный_код⟩} {⟨конечный_код⟩}
```

Создаёт окружение с заданным именем, подобное окружению `lstlisting`.

# Команда определения окружения

## Пример

```
\lstnewenvironment {pas} %  
{%  
  \lstset %  
  { %  
    language = Pascal, %  
    gobble = 2 %  
  } %  
} {}  
  
\begin{pas}  
  begin  
    WriteLn("Hello")  
  end.  
\end{pas}
```

```
begin  
  WriteLn("Hello")  
end.
```

[xcolor](#)

[graphicx](#)

[hyperref](#)

[listings](#)

Назначение

Вывод 1

Вывод 2

Вывод 3

Настройки

Окружение

**Пример 1**

Стили

Диапазон

Язык

Текст

Фон

Символы

Пример 2

[datatool 1](#)

[datatool 2](#)

# Команда определения стиля

[xcolor](#)

[graphicx](#)

[hyperref](#)

[listings](#)

Назначение

Вывод 1

Вывод 2

Вывод 3

Настройки

Окружение

Пример 1

**Стили**

Диапазон

Язык

Текст

Фон

Символы

Пример 2

[datatool 1](#)

[datatool 2](#)

**`\lstdefinestyle`** {⟨имя\_стиля⟩} {⟨параметры⟩}

Сохраняет множество настроек в стиле с заданным именем.

## Пример

```
\lstdefinestyle {pascal} %  
{ %  
  language = Pascal, %  
  gobble = 2 %  
} %
```

```
\begin{lstlisting}%  
  [style = pascal]  
  var X: Real;  
\end{lstlisting}
```

```
var X: Real;
```

# Основные параметры настройки диапазона печати

[xcolor](#)

[graphicx](#)

[hyperref](#)

[listings](#)

Назначение

Вывод 1

Вывод 2

Вывод 3

Настройки

Окружение

Пример 1

Стили

**Диапазон**

Язык

Текст

Фон

Символы

Пример 2

[datatool 1](#)

[datatool 2](#)

**linerange** = { $\langle \text{начало}_1 \rangle$  -  $\langle \text{конец}_1 \rangle$ ,  $\langle \text{начало}_2 \rangle$  -  $\langle \text{конец}_2 \rangle$   
...}

Выбирает диапазоны строк для печати.

**gobble** =  $\langle \text{количество} \rangle$

Пропускает заданное количество первых символов в начале каждой строки. Действует только для окружения `lstlisting` и подобных ему, не действует для команд `\lstinline` и `\lstinputlisting`.

# Основные параметры выбора языка и стиля

xcolor

graphicx

hyperref

listings

Назначение

Вывод 1

Вывод 2

Вывод 3

Настройки

Окружение

Пример 1

Стили

Диапазон

**Язык**

Текст

Фон

Символы

Пример 2

datatool 1

datatool 2

**style** =  $\langle \text{имя\_стиля} \rangle$

Устанавливает настройки, сохранённые в заданном стиле командой `\lstdefinestyle`.

**language** = [ $\langle \text{диалект} \rangle$ ]  $\langle \text{язык} \rangle$

Выбирает заданный язык с заданным диалектом. Пакет определяет множество языков, есть возможность создавать собственные языки и дополнять существующие.

**also language** = [ $\langle \text{диалект} \rangle$ ]  $\langle \text{язык} \rangle$

Добавляет настройки (ключевые слова и т. д.) из другого языка.

# Основные параметры определения внешнего вида текста

[xcolor](#)

[graphicx](#)

[hyperref](#)

[listings](#)

Назначение

Вывод 1

Вывод 2

Вывод 3

Настройки

Окружение

Пример 1

Стили

Диапазон

Язык

**Текст**

Фон

Символы

Пример 2

[datatool 1](#)

[datatool 2](#)

**basicstyle** =  $\langle \text{КОД} \rangle$  (по умолчанию:  $\{\}$ )

Определяет код, исполняемый в начале вывода фрагмента. Например,  $\{\small \ttfamily\}$ .

**identifierstyle** =  $\langle \text{КОД} \rangle$  (по умолчанию:  $\{\}$ )

**commentstyle** =  $\langle \text{КОД} \rangle$  (по умолчанию:  $\text{\itshape}$ )

**stringstyle** =  $\langle \text{КОД} \rangle$  (по умолчанию:  $\{\}$ )

Определяют внешний вид слов, не являющихся ключевыми, комментариев и строк соответственно.

**keywordstyle** =  $[\langle N^{\circ} \rangle] \langle \text{КОД} \rangle$  (по умолчанию:  $\text{\bfseries}$ )

Определяют внешний вид ключевых слов заданного уровня.

# Основные параметры определения внешнего вида фона и рамки

<a href="#">xcolor</a>
<a href="#">graphicx</a>
<a href="#">hyperref</a>
<a href="#">listings</a>
Назначение
Вывод 1
Вывод 2
Вывод 3
Настройки
Окружение
Пример 1
Стили
Диапазон
Язык
Текст
<b>Фон</b>
Символы
Пример 2
<a href="#">datatool 1</a>
<a href="#">datatool 2</a>

**frame** = none, lines, leftline, ... (по умолчанию: **none**)

Определяет вид рамки вокруг фрагментов.

**backgroundcolor** = *⟨команда цвета⟩*

**rulecolor** = *⟨команда цвета⟩*

**fillcolor** = *⟨команда цвета⟩*

**rulesepcolor** = *⟨команда цвета⟩*

Определяют команды цвета для фона, рамок, промежутка между текстом и рамкой, и промежутка между двойными рамками, соответственно. Например,  
`\color {blue!65!white}`.

# Основные параметры отображения символов

- [xcolor](#)
- [graphicx](#)
- [hyperref](#)
- [listings](#)
- Назначение
- Вывод 1
- Вывод 2
- Вывод 3
- Настройки
- Окружение
- Пример 1
- Стили
- Диапазон
- Язык
- Текст
- Фон
- Символы**
- Пример 2
- [datatool 1](#)
- [datatool 2](#)

**inputencoding** =  $\langle \text{кодировка} \rangle$  (по умолчанию: `{}`)

Определяет кодировку входного файла в команде `\lstinputlisting`.

**showspaces** = `true, false` (по умолчанию: **false**)

Определяет, будут ли пробелы в коде (кроме строк) отображаться в виде символа «`□`» или обычного пробела.

**showstringspaces** = `true, false` (по умолчанию: **true**)

Аналогично, для пробелов внутри строковых литералов.

**showtabs** = `true, false` (по умолчанию: **false**)

Аналогично, для символов табуляции во всём коде.

**tabsize** =  $\langle \text{число} \rangle$  (по умолчанию: 8)

Устанавливает позиции табуляции, кратные  $\langle \text{числу} \rangle$ .

# Пример отображения символов

[xcolor](#)

[graphicx](#)

[hyperref](#)

[listings](#)

Назначение

Вывод 1

Вывод 2

Вывод 3

Настройки

Окружение

Пример 1

Стили

Диапазон

Язык

Текст

Фон

Символы

**Пример 2**

[datatool 1](#)

[datatool 2](#)

## Пример

```
\begin{lstlisting}%  
  [language = Pascal, %  
  showspaces]  
begin  
  WriteLn('Hello world!')  
end.  
\end{lstlisting}  
\par \vspace {2em}  
\begin{lstlisting}%  
  [language = Pascal, %  
  showstringspaces = false]  
begin  
  WriteLn('Hello world!')  
end.  
\end{lstlisting}
```

```
__begin  
____WriteLn('Hello_world!')  
__end.  
  
begin  
  WriteLn('Hello world!')  
end.
```

xcolor

graphicx

hyperref

listings

**datatool 1**

Назначение

Загрузка 1

Загрузка 2

Обход 1

Обход 2

Обход 3

Проверки 1

Проверки 2

Проверки 3

Проверки 4

Проверки 5

Сортировка

datatool 2

# Пакет datatool

# Назначение пакета `datatool`

<u><code>xcolor</code></u>
<u><code>graphicx</code></u>
<u><code>hyperref</code></u>
<u><code>listings</code></u>
<u><code>datatool 1</code></u>
<b>Назначение</b>
Загрузка 1
Загрузка 2
Обход 1
Обход 2
Обход 3
Проверки 1
Проверки 2
Проверки 3
Проверки 4
Проверки 5
Сортировка
<u><code>datatool 2</code></u>

Пакет предназначен для обработки баз данных в виде таблиц.  
Функции пакета:

- ✓ Создание таблиц, добавление/удаление строк, установка текущей строки, редактирование данных в текущей строке.
- ✓ Загрузка/сохранение данных из/в текстовые файлы в формате CSV.
- ✓ Обход таблиц построчно с возможностью генерирования в выводе произвольного LaTeX-кода для данных каждой строки.
- ✓ Функции генерирования кода в зависимости от условий (сравнение чисел, строк, проверка на то, что строка является десятичной записью целого числа и т. д.)
- ✓ Визуализация данных таблиц в генерируемом документе в виде таблиц, диаграмм и т. д..

# Загрузка таблицы

- [xcolor](#)
- [graphicx](#)
- [hyperref](#)
- [listings](#)
- [datatool 1](#)
- Назначение
- Загрузка 1**
- Загрузка 2
- Обход 1
- Обход 2
- Обход 3
- Проверки 1
- Проверки 2
- Проверки 3
- Проверки 4
- Проверки 5
- Сортировка
- [datatool 2](#)

**\DTLloaddb** [*⟨параметры⟩*] {⟨имя\_базы⟩} {⟨путь\_к\_файлу⟩}

Создаёт базу данных (таблицу) с именем ⟨имя\_базы⟩ и загружает её содержимое из текстового файла по указанному пути. Файл должен иметь формат CSV и начинаться с необязательного заголовка, задающего имена колонок таблицы. Основные параметры:

**noheader** = *true, false* (по умолчанию: **false**)

Указывает, что первая строка файла не содержит заголовка.

**keys** ⟨*список\_ключей*⟩

Определяет имена (ключи) колонок (по умолчанию задаются в заголовке текстового файла).

# Загрузка таблицы (окончание)

<u>xcolor</u>
<u>graphicx</u>
<u>hyperref</u>
<u>listings</u>
<u>datatool 1</u>
Назначение
Загрузка 1
<b>Загрузка 2</b>
Обход 1
Обход 2
Обход 3
Проверки 1
Проверки 2
Проверки 3
Проверки 4
Проверки 5
Сортировка
<u>datatool 2</u>

## **\DTLsetseparator** {*СИМВОЛ*}

Устанавливает указанный *СИМВОЛ* в качестве символа-разделителя ячеек данных в строке таблицы при загрузке командой `\DTLloaddb`. По умолчанию используется запятая (« , »).

## **\DTLsetdelimiter** {*СИМВОЛ*}

Устанавливает *СИМВОЛ* в качестве символа-ограничителя данных одной ячейки. По умолчанию используется кавычка (« " »).

*Замечание:* кодировка загружаемого текстового файла должна совпадать с кодировкой `tex`-файла, который использует команду `\DTLloaddb`. Эта кодировка должна быть указана в качестве параметра пакета `inputenc`.



# Обход таблицы

- xcolor
- graphicx
- hyperref
- listings
- datatool 1
  - Назначение
  - Загрузка 1
  - Загрузка 2
  - Обход 1**
  - Обход 2
  - Обход 3
  - Проверки 1
  - Проверки 2
  - Проверки 3
  - Проверки 4
  - Проверки 5
  - Сортировка
- datatool 2

**\DTLforeach\*** [*⟨условие⟩*] {⟨имя\_базы⟩}  
{⟨список\_присваиваний⟩} {⟨текст⟩}

Генерирует вывод документа, повторяя заданный *⟨текст⟩* для каждой строки таблицы с заданным именем (*⟨имя\_базы⟩*). *⟨список\_присваиваний⟩* содержит определения макросов, которые внутри текста заменяются данными текущей строки. Список содержит разделяемые запятыми присваивания:

*⟨имя\_команды⟩*=*⟨ключ⟩*

(без пробелов), где *⟨ключ⟩* является именем колонки (см. определение `\DTLloaddb`). Необязательное *⟨условие⟩* определяет, следует ли обрабатывать текущую строку.

# Обход таблицы (продолжение)

- [xcolor](#)
- [graphicx](#)
- [hyperref](#)
- [listings](#)
- [datatool 1](#)
- Назначение
- Загрузка 1
- Загрузка 2
- Обход 1
- Обход 2**
- Обход 3
- Проверки 1
- Проверки 2
- Проверки 3
- Проверки 4
- Проверки 5
- Сортировка
- [datatool 2](#)

## Замечания:

- ✓ Вариант команды `\DTLforeach` «со звёздочкой» не способен обеспечить изменения таблицы во время её обхода, однако работает эффективнее версии «без звёздочки». Поэтому в большинстве случаев он предпочтительнее.
- ✓ Необязательные условия для отбора строк таблицы могут состоять из команд проверки (см. далее), а также команд пакета `ifthen` (`\and`, `\or`, `\not` и т. д.)



## Обход таблицы (окончание)

<u>xcolor</u>
<u>graphicx</u>
<u>hyperref</u>
<u>listings</u>
<u>datatool 1</u>
Назначение
Загрузка 1
Загрузка 2
Обход 1
Обход 2
<b>Обход 3</b>
Проверки 1
Проверки 2
Проверки 3
Проверки 4
Проверки 5
Сортировка
<u>datatool 2</u>

```
\begin{DTLenvforeach}[\langleусловие\rangle] {\langleимя_базы\rangle}
{\langleсписок_присваиваний\rangle}
{\langleсодержимое_окружения\rangle}
\end{DTLenvforeach}
```

```
\begin{DTLenvforeach*}[\langleусловие\rangle] {\langleимя_базы\rangle}
{\langleсписок_присваиваний\rangle}
{\langleсодержимое_окружения\rangle}
\end{DTLenvforeach*}
```

Варианты команд `\DTLforeach` и `\DTLforeach*` в виде окружений. `\langleсодержимое_окружения\rangle` используется вместо параметра `\langleтекст\rangle` указанных команд.

# Проверка условий

*Замечание:* для удобства использования некоторые из команд проверки условий существуют в двух вариантах:

для использования внутри условия команды **\ifthenelse** пакета `ifthen` (`\DTLis...`) и отдельно от неё (`\DTLif...`). Команды последнего типа получают два дополнительных параметра: то, чем они заменяются в случае истинности условия, и в случае ложности.



## Пример

```
\newcommand {\num} {3}
\DTLifint {\num} %
  {\num\ is integer} {\num isn't integer} \
\ifthenelse %
  {\DTLisint {\num} \and \DTLisgt {\num} {5}} %
  {$\num > 5$} {$\num \le 5$}
```

3 is integer  
3 ≤ 5

- xcolor
- graphicx
- hyperref
- listings
- datatool 1
- Назначение
- Загрузка 1
- Загрузка 2
- Обход 1
- Обход 2
- Обход 3
- Проверки 1**
- Проверки 2
- Проверки 3
- Проверки 4
- Проверки 5
- Сортировка
- datatool 2

# Проверка условий (продолжение)

<a href="#">xcolor</a>
<a href="#">graphicx</a>
<a href="#">hyperref</a>
<a href="#">listings</a>
<a href="#">datatool 1</a>
Назначение
Загрузка 1
Загрузка 2
Обход 1
Обход 2
Обход 3
Проверки 1
<b>Проверки 2</b>
Проверки 3
Проверки 4
Проверки 5
Сортировка
<a href="#">datatool 2</a>

**\DTLifint** {⟨текст⟩} {⟨если\_истина⟩} {⟨если\_ложь⟩}

**\DTLifreal** {⟨текст⟩} {⟨если\_истина⟩} {⟨если\_ложь⟩}

**\DTLifcurrency** {⟨текст⟩} {⟨если\_истина⟩} {⟨если\_ложь⟩}

Проверяют, является ли ⟨текст⟩ записью, соответственно, целого, вещественного числа или денежной суммы (`\$5.99`, `\pounds 5.99`, `\texteuro 5.99`).

**\DTLifnumerical** {⟨текст⟩} {⟨если\_истина⟩} {⟨если\_ложь⟩}

Проверяет, является ли ⟨текст⟩ записью целого или вещественного числа или денежной суммы.

**\DTLifstring** {⟨текст⟩} {⟨если\_истина⟩} {⟨если\_ложь⟩}

Противоположно `\DTLifnumerical`.

## Проверка условий (продолжение)

<a href="#">xcolor</a>
<a href="#">graphicx</a>
<a href="#">hyperref</a>
<a href="#">listings</a>
<a href="#">datatool 1</a>
Назначение
Загрузка 1
Загрузка 2
Обход 1
Обход 2
Обход 3
Проверки 1
Проверки 2
<b>Проверки 3</b>
Проверки 4
Проверки 5
Сортировка
<a href="#">datatool 2</a>

**\DTLifnum** $\langle\text{операция}\rangle$   $\{\langle\text{число}_1\rangle\}$   $\{\langle\text{число}_2\rangle\}$   $\{\langle\text{если\_истина}\rangle\}$   
 $\{\langle\text{если\_ложь}\rangle\}$

Проверяет числа на равенство (если суффикс  $\langle\text{операция}\rangle$  равен eq), отношение «меньше» (lt) и «больше» (gt).

**\DTLifstring** $\langle\text{операция}\rangle^*$   $\{\langle\text{строка}_1\rangle\}$   $\{\langle\text{строка}_2\rangle\}$   
 $\{\langle\text{если\_истина}\rangle\}$   $\{\langle\text{если\_ложь}\rangle\}$

Проверяет строки на лексикографическое равенство и т. д. Версия «со звёздочкой» выполняет сравнение без учёта регистра.

**\DTLif** $\langle\text{операция}\rangle^*$   $\{\langle\text{аргумент}_1\rangle\}$   $\{\langle\text{аргумент}_2\rangle\}$   
 $\{\langle\text{если\_истина}\rangle\}$   $\{\langle\text{если\_ложь}\rangle\}$

Если  $\langle\text{аргумент}_1\rangle$  и  $\langle\text{аргумент}_2\rangle$  являются числами, эквивалентна  $\backslash\text{DTLifnum}\langle\text{операция}\rangle$ , иначе —  $\backslash\text{DTLifstring}\langle\text{операция}\rangle^*$ .

# Проверка условий (продолжение)

[xcolor](#)

[graphicx](#)

[hyperref](#)

[listings](#)

[datatool 1](#)

Назначение

Загрузка 1

Загрузка 2

Обход 1

Обход 2

Обход 3

Проверки 1

Проверки 2

Проверки 3

**Проверки 4**

Проверки 5

Сортировка

[datatool 2](#)

**\DTLisint** {⟨текст⟩}

**\DTLisreal** {⟨текст⟩}

**\DTLiscurrency** {⟨текст⟩}

**\DTLisnumerical** {⟨текст⟩}

**\DTLisstring** {⟨текст⟩}

**\DTLis**⟨операция⟩ {⟨аргумент<sub>1</sub>⟩} {⟨аргумент<sub>2</sub>⟩}

Варианты приведённых ранее команд для использования внутри условия команды **\ifthenelse**. ⟨операция⟩ должна быть одной из: eq, lt, gt, а также ieq, ilt, igt (без учёта регистра).

# Проверка условий (окончание)

*Замечание:* следующие команды можно использовать только в аргументе  $\langle \text{текст} \rangle$  команды `\DTLforeach` или внутри окружения `DTLenvforeach`, но не в их аргументе  $\langle \text{условие} \rangle$ . △

## **\DTLcurrentindex**

Разворачивается в десятичную запись номера текущей строки таблицы.

## **\DTLiffirstrow** $\{\langle \text{если\_истина} \rangle\} \{\langle \text{если\_ложь} \rangle\}$

Проверяет, является ли текущая строка первой.

## **\DTLiflastrow** $\{\langle \text{если\_истина} \rangle\} \{\langle \text{если\_ложь} \rangle\}$

Проверяет, является ли текущая строка последней.

## **\DTLifoddrrow** $\{\langle \text{если\_истина} \rangle\} \{\langle \text{если\_ложь} \rangle\}$

Проверяет, является ли номер текущей строки нечётным.

<u>xcolor</u>
<u>graphicx</u>
<u>hyperref</u>
<u>listings</u>
<u>datatool 1</u>
Назначение
Загрузка 1
Загрузка 2
Обход 1
Обход 2
Обход 3
Проверки 1
Проверки 2
Проверки 3
Проверки 4
<b>Проверки 5</b>
Сортировка
<u>datatool 2</u>

# Сортировка таблицы

<a href="#">xcolor</a>
<a href="#">graphicx</a>
<a href="#">hyperref</a>
<a href="#">listings</a>
<a href="#">datatool 1</a>
Назначение
Загрузка 1
Загрузка 2
Обход 1
Обход 2
Обход 3
Проверки 1
Проверки 2
Проверки 3
Проверки 4
Проверки 5
<b>Сортировка</b>
<a href="#">datatool 2</a>

**\DTLsort\*** {⟨критерии\_сортировки⟩} {⟨имя\_базы⟩}

Сортирует таблицу с именем {⟨имя\_базы⟩} в соответствии с заданными критериями. Версия команды «со звёздочкой» использует сравнение без учёта регистра. {⟨критерии\_сортировки⟩} содержит разделённый запятыми (без пробелов) список имён колонок, по которым необходима сортировка. Для сортировки по какой-либо из колонок в обратном порядке необходимо указать после её имени строку «=descending».

xcolor

graphicx

hyperref

listings

datatool 1

**datatool 2**

Пример 1

Пример 2

Пример 3

Пример 4

Пример 5

Пример 6

Пример 7

Пример 8

Пример 9

\let

## Пакет datatool (окончание)

# Пример генерирования документа на основе базы данных

Пусть информация о студентах (фамилия, имя, отчество, курс, группа, рейтинг) хранится в текстовом файле следующего вида:

Пример (students.txt)

```
Иванов Иван Иванович 2 1 87
Петров Пётр Петрович 2 1 76
Сидоров Сидор Сидорович 2 2 64
```

Требуется сначала вывести списки студентов каждой группы в соответствующих подразделах документа, упорядоченные по ФИО, затем — в одном списке, упорядоченном по рейтингу в порядке убывания, курсу, группе, ФИО.

- [xcolor](#)
- [graphicx](#)
- [hyperref](#)
- [listings](#)
- [datatool 1](#)
- [datatool 2](#)
- [Пример 1](#)**
- [Пример 2](#)
- [Пример 3](#)
- [Пример 4](#)
- [Пример 5](#)
- [Пример 6](#)
- [Пример 7](#)
- [Пример 8](#)
- [Пример 9](#)
- [\let](#)

# Пример генерирования документа на основе базы данных (продолжение)

[xcolor](#)

[graphicx](#)

[hyperref](#)

[listings](#)

[datatool 1](#)

[datatool 2](#)

[Пример 1](#)

**[Пример 2](#)**

[Пример 3](#)

[Пример 4](#)

[Пример 5](#)

[Пример 6](#)

[Пример 7](#)

[Пример 8](#)

[Пример 9](#)

[\let](#)

```
\documentclass [a4paper, 12pt] {article}
```

```
\usepackage [T1] {fontenc}
```

```
\usepackage [cp1251] {inputenc}
```

```
\usepackage [russian] {babel}
```

```
\usepackage {datatool}
```

```
\begin {document}
```

```
% Установка пробела как разделителя между ячейками
```

```
\DTLsetseparator { }
```

# Пример генерирования документа на основе базы данных (продолжение)

[xcolor](#)

[graphicx](#)

[hyperref](#)

[listings](#)

[datatool 1](#)

[datatool 2](#)

[Пример 1](#)

[Пример 2](#)

**[Пример 3](#)**

[Пример 4](#)

[Пример 5](#)

[Пример 6](#)

[Пример 7](#)

[Пример 8](#)

[Пример 9](#)

[\let](#)

```
% Загрузка таблицы
```

```
\DTLloaddb %
```

```
[ %
```

```
noheader, %
```

```
keys = {Surname,FirstName,ThirdName,Course,Group,Rating} %
```

```
] %
```

```
{students} %
```

```
{students.txt}
```

```
\section* {Группы}
```

```
% Начальные значения номера курса и группы: 0
```

```
\newcommand {\oldcourse} {0}
```

```
\newcommand {\oldgroup} {0}
```

# Пример генерирования документа на основе базы данных (продолжение)

[xcolor](#)

[graphicx](#)

[hyperref](#)

[listings](#)

[datatool 1](#)

[datatool 2](#)

[Пример 1](#)

[Пример 2](#)

[Пример 3](#)

[Пример 4](#)

[Пример 5](#)

[Пример 6](#)

[Пример 7](#)

[Пример 8](#)

[Пример 9](#)

[\let](#)

```
% Сортировка таблицы по курсу, группе, ФИО
\DTLsort* %
  {Course,Group,Surname,FirstName,ThirdName} %
  {students}
```

# Пример генерирования документа на основе базы данных (продолжение)

xcolor

graphicx

hyperref

listings

datatool 1

datatool 2

Пример 1

Пример 2

Пример 3

Пример 4

**Пример 5**

Пример 6

Пример 7

Пример 8

Пример 9

\let

```
% Обработка строк таблицы
```

```
\begin {DTLenvforeach*} %
```

```
[ %
```

```
% Условие отбора: номера курса, группы и рейтинга - целые числа
```

```
\DTLisint {\course} \and %
```

```
\DTLisint {\group} \and %
```

```
\DTLisint {\rating} %
```

```
] %
```

```
{students} %
```

```
{ %
```

```
\surname=Surname, %
```

```
\firstname=FirstName, %
```

```
\thirdname=ThirdName, %
```

```
\course=Course, %
```

```
\group=Group, %
```

```
\rating=Rating%
```

```
} %
```

# Пример генерирования документа на основе базы данных (продолжение)

xcolor

graphicx

hyperref

listings

datatool 1

datatool 2

Пример 1

Пример 2

Пример 3

Пример 4

Пример 5

**Пример 6**

Пример 7

Пример 8

Пример 9

\let

```
%  
% Обработка одной строки  
\ifthenelse %  
{ %  
  % Номера курса и группы совпадают с ними на предыдущей итерации?  
  \DTLiseq {\course} {\oldcourse} \and %  
  \DTLiseq {\group} {\oldgroup}%  
} %  
{ %  
  % Если да, то сделать в начале перевод строки  
  \\  
} %
```

# Пример генерирования документа на основе базы данных (продолжение)

[xcolor](#)

[graphicx](#)

[hyperref](#)

[listings](#)

[datatool 1](#)

[datatool 2](#)

[Пример 1](#)

[Пример 2](#)

[Пример 3](#)

[Пример 4](#)

[Пример 5](#)

[Пример 6](#)

**[Пример 7](#)**

[Пример 8](#)

[Пример 9](#)

[\let](#)

```
{ %  
  % Иначе вывести команду подраздела  
  \subsection* {\course.\group}  
  %  
  % Присвоить предыдущим значениям текущие  
  \let \oldcourse \course %  
  \let \oldgroup \group %  
  %  
} %  
%  
% Вывести фамилию, имя и отчество  
\surname\ \firstname\ \thirdname  
%  
\end {DTLenvforeach*}
```

# Пример генерирования документа на основе базы данных (продолжение)

[xcolor](#)

[graphicx](#)

[hyperref](#)

[listings](#)

[datatool 1](#)

[datatool 2](#)

[Пример 1](#)

[Пример 2](#)

[Пример 3](#)

[Пример 4](#)

[Пример 5](#)

[Пример 6](#)

[Пример 7](#)

**[Пример 8](#)**

[Пример 9](#)

[\let](#)

```
\section* {Рейтинг}

% Сортировка таблицы по рейтингу, курсу и т. д.
\DTLsort* %
  {Rating=descending, Course, Group, Surname, FirstName, ThirdName} %
  {students}

% Обработка строк таблицы
\begin {DTLenvforeach*} %
  {students} %
  { %
    \rating=Rating, %
    \surname=Surname, %
    \firstname=FirstName, %
    \thirdname=ThirdName%
  } %
```

# Пример генерирования документа на основе базы данных (окончание)

xcolor

graphicx

hyperref

listings

datatool 1

datatool 2

Пример 1

Пример 2

Пример 3

Пример 4

Пример 5

Пример 6

Пример 7

Пример 8

**Пример 9**

`\let`

```
%  
% Обработка одной строки  
% Вывести фамилию, имя, отчество и рейтинг  
\surname\ \firstname\ \thirdname\ \rating %  
% Если строка в таблице не последняя,  
%   в конце выполнить перевод строки  
\DTLiflastrow {} {\}\}  
%  
\end {DTLenvforeach*}  
  
\end {document}
```

# Команда `\let`

Использованная в примере команда `\let` является примитивом TeX, при помощи которого можно скопировать содержимое одной команды в другую:

```
\let <имя_команды-приёмника> <имя_команды-источника>
```

*Замечание:* использование команды `\renewcommand` вместо `\let` в примере выше невозможно из-за того, что `\renewcommand` (пере)определяет команду (`\oldcourse`) в терминах другой команды (`\course`), однако команда `\course` меняется на следующей итерации, из-за чего возвращаемое значение `\oldcourse` также меняется.



- [xcolor](#)
- [graphicx](#)
- [hyperref](#)
- [listings](#)
- [datatool 1](#)
- [datatool 2](#)
- [Пример 1](#)
- [Пример 2](#)
- [Пример 3](#)
- [Пример 4](#)
- [Пример 5](#)
- [Пример 6](#)
- [Пример 7](#)
- [Пример 8](#)
- [Пример 9](#)
- [\let](#)**