

Включение рисунков в $\text{\LaTeX} 2_{\epsilon}^*$

Описаны команды, введённые в пакете `graphicx`, входящего в коллекцию `graphics`. Пакет позволяет включать в документ рисунки из графических файлов, поворачивать, растягивать или сжимать боксы. Описан также пакет `lscapex` из этой же коллекции.

Содержание

1	Выбор драйвера	2
2	Выбор формата рисунка	2
2.1	Драйвер <code>dvips</code>	2
2.1.1	Преобразование JPEG в EPS	2
2.1.2	Преобразование TIFF в EPS	3
2.2	Драйвер <code>pdftex</code>	3
3	Вставка рисунка из графического файла	3
3.1	Как задать размер рисунка в документе	4
3.2	Включение в документ части рисунка	5
3.3	Поворот рисунка	6
3.4	Имена файлов без расширения	6
3.4.1	Нестандартные расширения	7
3.5	Местоположение файлов с рисунками	7
3.6	«Нестандартные» файлы с рисунком	7
3.7	Черновой режим	9
4	Манипуляции с боксами	9
4.1	Изменение размеров бокса	9
4.1.1	Трансформация к указанному размеру	9
4.1.2	Трансформация по указанному масштабу	10
4.2	Зеркальное отражение блока	10
4.3	Поворот блока	10
5	Альбомная ориентация страницы	11
6	Общие настройки графического пакета	11
6.1	Опции пакета	11
6.2	Установка ключей	11
	Алфавитный указатель	12

*© 2001 Владимир Сюткин. Замечания приветствуются: syutkin@ns.kinetics.nsc.ru

1 Выбор драйвера

При загрузки пакетов надо в необязательном аргументе команды `\usepackage` указать драйвер или, другими словами, программу, которая «понимает» команды из этих пакетов. Драйвером по умолчанию задан `dvips`¹. Программа `dvips`, которую написал Tomas Rokicki, переводит `dvi`-файл, полученный после обработки `LATEX`'ом входного файла, в файл, в котором документ описан на языке PostScript. `ps`-файл можно с помощью программы `GhostScript`² просмотреть на экране монитора и распечатать на обычном принтере. Кроме того, `GhostScript` умеет конвертировать `ps`-файл в файл формата PDF. Если же вы создаёте документ в формате PDF непосредственно с помощью `pdfLATEX`'а, написанного Hàn Thê Thành, то вам следует при загрузки пакетов указывать в виде опции драйвер `pdftex`.

Остальные драйверы, которые в настоящее время декларированы для пакетов из коллекции `graphics`: `xdvi`, `dvipdf`, `dvipdfm`, `dvipsone`, `dviwindo`, `emtex`, `dviwin`, `pctexps`, `pctexwin`, `pctexhp`, `pctex32`, `truetetex`, `tcidvi`, `vtex`, `oztex`, `textures`.

2 Выбор формата рисунка

2.1 Драйвер `dvips`

Драйвер `dvips` поддерживает включение в документ рисунков только из файлов в формате EPS (Encapsulated PostScript). `eps`-файл имеет в обязательном порядке строку:

```
%%BoundingBox: llx lly urx ury
```

где целые числа `llx`, `lly`, `urx` и `ury` — это *x*- и *y*-координаты в больших пунктах³ левого нижнего и правого верхнего углов области, внутри которой находится рисунок на «воображаемой» странице. Именно эта часть страницы импортируется в документ.

В наших примерах будет использоваться файл `a.eps`, в котором область с рисунком задана как:

```
%%BoundingBox: 14 14 88 88
```

Следовательно, точка отсчёта рисунка (левый нижний угол) находится на расстоянии 14 bp как от левого, так и от нижнего краёв страницы, а сам рисунок имеет размер 72 bp на 72 bp или примерно 2.54 см на 2.54 см.

Подчеркнём, что сам `LATEX` читает из `eps`-файла только параметры `BoundingBox` для того, чтобы знать, сколько места на странице надо отвести под рисунок. Рисунок из `eps`-файла читает и вставляет в документ драйвер `dvips`, когда переводит DVI в PS.

Если необходимо вставить в документ растровый рисунок, то надо сначала преобразовать файл с этим рисунком в `eps`-файл. Многие графические редакторы и программы сохраняют растровую графику в виде `eps`-файла. Мы обсудим здесь только две свободные программы, которые обычно входят в поставку `LATEX`'а и делают эту работу лучше любой программы, включая даже `GhostScript`.

2.1.1 Преобразование JPEG в EPS

JPEG используется для растровых рисунков с плавными переходами от одного цвета к другому, например, для фотографических изображений. Только в этом случае он даёт эффективное сжатие данных практически без потери качества для зрительного восприятия.

Программа `jpeg2ps.exe` преобразует файл в формате JPEG в `eps`-файл. При этом сам рисунок не конвертируется в формат EPS. `jpeg2ps` записывает в заголовок `eps`-файла информацию о параметрах рисунка, а затем копирует туда JPEG-данные. Распаковка данных осуществляется интерпретатором PostScript'a, например, программой `GhostScript`, на этапе просмотра или печати

¹Драйвер по умолчанию задаётся в аргументе команды `\ExecuteOptions` в файле настройки `graphics.cfg`.

²`GhostScript` является интерпретатором языка PostScript и используется в среде Windows под оболочкой `GSview`.

³72 больших пункта (bp) равны 1 дюйму или 2.54 см

рисунка. Для преобразования jpg-файла в eps-файл достаточно выполнить следующую командную строку:

```
jpeg2ps -r 0 -o outputfile.eps inputfile.jpg
```

Опция `-o` переводит программу `jpeg2ps` в режим записи выходных данных в файл на диске. Опция `-r 0` задаёт размер Bounding Box, равным размеру рисунка в пикселах. Так, если размер рисунка равен 600 px на 300 px, то размер Bounding Box будет 600 bp на 300 bp. При этом координаты точки отсчёта обычно не равны (0,0). В общем случае опция `-r` имеет вид `-r dpi`, где *dpi* — число точек на дюйм. Так, если в случае рисунка размером 600 px на 300 px задать опцию в виде `-r 600`, то размер Bounding Box будет 72 bp на 36 bp или 1 дюйм (600/600) на 0.5 дюйма (300/600).

2.1.2 Преобразование TIFF в EPS

TIFF является одним из самых подходящих форматов для растровых рисунков с резкими границами между цветами, например, для схем, диаграмм или графиков.

Программа `tiff2ps.ps` из набора `pstools` с помощью GhostScript'a преобразует файл в формате TIFF в eps-файл. Формат TIFF позволяет хранить рисунки в сжатом виде (обычно используется метод LZW). eps-файл, полученный из такого tif-файла, может иметь достаточно большой размер. Программа `tiff2ps.ps` позволяет избежать хранения одного и того же рисунка сразу в двух файлах. Если использовать опцию `-h`, то в заголовке eps-файла будет записана ссылка на tif-файл, а сам рисунок не будет копироваться туда. Вот типичный пример такой командной строки (`gswin32` — это GhostScript в среде Windows):

```
gswin32c -- tiff2ps.ps inputfile.tif outputfile.eps -i -h -b0
```

2.2 Драйвер pdftex

pdfL^AT_EX не поддерживает формат EPS. Зато он позволяет включать в документ графику в векторном формате PDF. Преобразовать рисунок из EPS в PDF можно, например, с помощью GhostScript'a прямо из GSview. Но обычно конверторы не задают в pdf-файле параметры CropBox'a (аналог BoundingBox формата EPS). Поэтому pdfT_EX при включении рисунка в документ использует параметры MediaBox'a, задающие размер листа с рисунком, а не размер самого рисунка. Чтобы получить правильный Bounding Box из eps-файла, надо перед конвертированием преобразовать eps-файл так, чтобы точкой отсчёта стала точка (0,0) и чтобы размер страницы точно соответствовал BoundingBox. Такую работу делает программа `epstopdf.exe`. В командной строке надо задавать только имена входного eps-файла и выходного pdf-файла:

```
epstopdf inputfile.eps outputfile.pdf
```

Так, в файле `a.pdf`, полученном из файла `a.eps` с помощью `epstopdf`, MediaBox задан как `[0 0 72 72]`.

Драйвер `pdftex`, в отличие от `dvips`, поддерживает включение в документ рисунков в растровых форматах PNG, TIFF и, что особенно важно для фотографических изображений, в формате JPEG.

3 Вставка рисунка из графического файла

В пакете `graphicsx` определена команда

```
\includegraphics[keyval-list]{file}
```

для вставки в документ рисунка из графического файла *file*. Необязательный аргумент *keyval-list* может содержать целый список ключей. Значения ключей задаются в виде *key=value*, а в списке они перечисляются через запятую. Команда `\includegraphics` не заканчивает абзац, поэтому позволяет вставлять небольшие рисунки прямо внутрь текста.

Опцию команды `\includegraphics` можно опустить, если мы хотим вставить в документ рисунок в «натуральную» величину из eps-файла в случае драйвера `dvips` или из pdf-файла в случае драйвера `pdftex`. \LaTeX выделяет в документе под рисунок бокс, размер которого считывается из `BoundingBox` и `MediaBox`, соответственно. Пример⁴:

```
\includegraphics{a}
```



Видно, что точкой отсчёта бокса, выделенного под рисунок, является его левый нижний угол. В этом примере расширение файла с рисунком опущено. Это сделано для того, чтобы входной файл можно было обрабатывать как \LaTeX 'ом, так и `pdf \LaTeX` 'ом: \LaTeX берёт рисунок из файла `a.eps`, а `pdf \LaTeX` — из файла `a.pdf`. Более подробно читайте об этом в разделе 3.4.

На заметку Один и тот же входной файл можно обрабатывать как \LaTeX 'ом, так и `pdf \LaTeX` 'ом, если воспользоваться конструкцией из следующего примера:

```
\ifx\pdfoutput\undefined
\usepackage{graphicx}
\else
\usepackage[pdftex]{graphicx}
\fi
```

`BoundingBox` в eps-файле задаёт размер рисунка с точностью до 1 bp. Бывают случаи, когда такой точности недостаточно. Тогда можно использовать параметры из строки (если она присутствует в файле)

```
%\HiResBoundingBox: llx lly urx ury
```

в которой числа с десятичной точкой `llx`, `lly`, `urx` и `ury` (они имеют тот же смысл, что и для `BoundingBox`) задают размер рисунка с более высокой точностью, чем 1 bp. Для того, чтобы \TeX читал именно эту строку, надо в команде `\includegraphics` указать ключ

```
hiresbb
```

3.1 Как задать размер рисунка в документе

В этом разделе описаны ключи команды `\includegraphics`, с помощью которых можно задать размер прямоугольной области, выделяемой для размещения рисунка в документе.

Ключ

```
width=length
```

устанавливает значение `length` (в любых \TeX 'овских единицах длины) в качестве ширины области, выделяемой для размещения рисунка. Пример:

```
\includegraphics[width=1.5cm]{a}
```



Ключи

```
height=length
totalheight=length
```

⁴Кроме рисунка (он выполнен синим цветом), в этом и следующих примерах показаны также базовая линия строки с рисунком и границы бокса, отведённого под рисунок.

устанавливают значение *length* (в любых Т_ЕX’овских единицах длины) в качестве высоты (полной высоты) области, выделяемой для размещения рисунка. Пример:

```
\includegraphics[width=1in,height=10mm]{a}
```



Ключ

```
keepaspectratio
```

обеспечивает сохранение отношения ширины к высоте самого рисунка, если заданные значения ширины и высоты области, выделенной под рисунок, нарушают, как в предыдущем примере, это отношение. Пример:

```
\includegraphics[width=1in,height=1cm,%  
keepaspectratio]{a}
```



Видно, что в этом примере размер бокса, отведённого под рисунок, задаётся значением ключа *height*. Ключ *width* просто игнорируется: иначе рисунок вышел бы за пределы выделенной под него области по вертикали.

Ключ

```
scale=scale
```

изменяет «натуральный» размер рисунка в *scale* раз. Пример:

```
\includegraphics[scale=0.5]{a}
```



Результат, аналогичный действию описанных выше ключей, можно получить с помощью команд изменения размеров боксов, описанных в разделах [4.1.1](#) и [4.1.2](#).

Для полноты изложения материала надо упомянуть об «устаревшем» ключе для указания размера бокса, который надо выделить под рисунок:

```
bb=llx lly urx ury
```

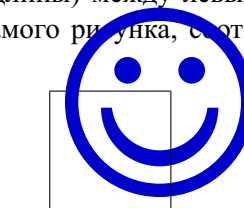
3.2 Включение в документ части рисунка

Ключи

```
viewport=llx lly urx ury  
trim=dl db dr du
```

задают так называемую видимую область рисунка. Именно под эту часть рисунка отводится место в документе. Здесь *llx*, *lly*, *urx* и *ury* — это x- и y-координаты (в любых Т_ЕX’овских единицах длины) левого нижнего и правого верхнего углов видимой области рисунка относительно точки отсчёта, а *dl*, *db*, *dr* и *du* — это расстояния (в любых Т_ЕX’овских единицах длины) между левыми, нижними, правыми и верхними границами видимой области рисунка и самого рисунка, соответственно. Отрицательные значения смещения допустимы. Пример:

```
\includegraphics[viewport=-5 -5 40 40]{a}
```



Ключ

```
clip=boolean
```

отсекает часть рисунка, выходящую за границы видимой области, если значение *boolean* равно `true` (или просто указано). Пример:

```
\includegraphics[trim=-5 -5 16 16,clip]{a}
```



3.3 Поворот рисунка

Ключ

```
angle=angle
```

поворачивает рисунок на *angle* градусов против часовой стрелки. По умолчанию ось вращения проходит через точку отсчёта бокса. Пример:

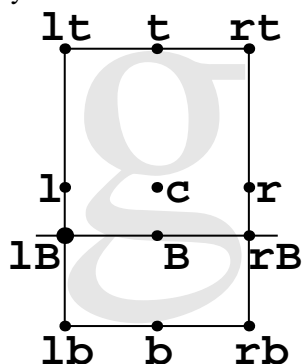
```
\includegraphics[scale=0.6,angle=30]{a}
```



Ключ

```
origin=pos
```

позволяет указать одно из 12 предопределённых положений оси вращения, показанных на следующем рисунке:



Положение оси по горизонтали задаётся одной из трёх букв: *l* (на левой стороне бокса), *c* (по центру бокса) или *r* (на правой стороне бокса). Положение оси вращения по вертикали задаётся одной из четырёх букв: *b* (на нижней стороне бокса), *B* (на базовой линии), *c* (по центру бокса) или *t* (на верхней стороне бокса). В результате получается следующий список допустимых значений *pos*: *lb*, *lB*, *l*, *lt*, *b*, *B*, *c*, *t*, *rb*, *rB*, *r*, *rt*. Если указана только одна буква, то второй предполагается *c*. Так, `origin=lb` соответствует вращению вокруг точки отсчёта, в `origin=c` — вокруг центра бокса.

Положения точек вращения и допустимые значения *pos*.

Результат, аналогичный действию описанных в этом разделе ключей, можно получить с помощью команды вращения боксов, описанной в разделе 4.3.

На память Порядок следования ключей поворота и установки нового размера бокса через исходные размеры бокса влияет на результат, поскольку поворот меняет такие параметры бокса, как `\width` и др. (см. пример на странице 10). Ключи читаются слева направо!

3.4 Имена файлов без расширения

Расширение файла с рисунком в команде `\includegraphics` можно не указывать, поскольку драйвер сам знает, какие типы файлов он может обработать, а какие нет. Для драйвера `dvips` это файлы с расширением

eps, ps, eps.gz, ps.gz, eps.Z

а для драйвера pdftex

png, pdf, jpg, mps, tif

Когда расширение файла в команде `\includegraphics` не указано, драйвер последовательно (слева направо по списку) добавляет к имени файла все известные ему расширения и ищет файл уже по полному имени. Поиск прекращается, как только находится первый подходящий файл.

Пакет `graphicx` вводит декларацию

```
\DeclareGraphicsExtensions{ext-list}
```

которая позволяет вместо списка по умолчанию задать свой собственный список расширений, которые драйвер будет добавлять к имени файла во время его поиска. В *ext-list* расширения перечисляются через запятую, причем перед названием расширения ставится точка. Расширения используются в порядке их перечисления в списке.

На заметку Опуская в команде `\includegraphics` расширение файла с рисунком, можно один и тот же входной файл обрабатывать как \LaTeX 'ом, так и `pdf\LaTeX`'ом.

3.4.1 Нестандартные расширения

На заметку Драйвер `dvips` трактует все файлы с неизвестными ему расширениями как рисунки типа `eps`. Поэтому вы можете использовать для своих `eps`-файлов *любые* нестандартные расширения, например, `xyz`.

3.5 Местоположение файлов с рисунками

По умолчанию \LaTeX ищет файлы с рисунками в каталоге, в котором находится входной файл. Декларация

```
\graphicspath{dir-list}
```

позволяет расширить область поиска. В списке *dir-list* каждая из директорий заключается в фигурные скобки. Если указать, например

```
\graphicspath{{images/}{d:/images/eps/}{d:/images/pdf/}}
```

то \LaTeX будет искать файлы с рисунками также в подкаталоге `images` текущего каталога (где находится входной файл) и в директориях `d:/images/eps` и `d:/images/pdf`.

На заметку Поскольку `eps`-файл является текстовым файлом, мы можем хранить его прямо во входном файле в окружении `filecontents*` из стандартного \LaTeX 'а.

3.6 «Нестандартные» файлы с рисунком

На память Всё, описанное в этом разделе, относится только к драйверу `dvips` и *не поддерживается* драйвером `pdfTeX`.

Драйвер `dvips` умеет включать в документ рисунки из `eps`-файлов, которые сжаты программой `gzip`. Они распознаются по расширению `eps.gz`. Вот как выглядит предписание о трактовке файлов с таким расширением в файле `dvips.def`:

```
\@namedef{Gin@rule@.eps.gz}#1{{eps}{.eps.bb}{`gunzip -c #1}}
```

Здесь #1 заменяет собой имя файла *name*. Итак, в предписании сказано, что файл с расширением `eps.gz` содержит рисунок типа `eps`, размер рисунка надо читать в файле `name.eps.bb`, а перед включением рисунка в документ надо сначала распаковать файл `name.eps.gz` программой `gunzip`. Файл `name.eps.bb` с размером рисунка нужен \TeX 'у на стадии вёрстки документа. Он должен состоять из строки

```
%%BoundingBox: llx lly urx ury
```

взятой из `eps`-файла (её смысл описан на странице 2). Дело в том, что сам файл с рисунком сжат и \LaTeX не может прочитать из него размер рисунка.

С помощью декларации

```
\DeclareGraphicsRule{ext}{type}{read-file}{command}
```

можно: (i) связать расширение *ext* с типом рисунка *type*, понятным драйверу (`eps` для `dvips`); (ii) указать расширение файла *read-file*, из которого драйвер может получить данные о размере рисунка; (iii) задать команду *command*, которая должна быть выполнена перед включением рисунка в документ. Аргументы декларации `\DeclareGraphicsRule` имеют тот же смысл, что и в приведенной выше инструкции для расширения `eps.gz` из файла `dvips.def`. Перепишем эту инструкцию, используя декларацию `\DeclareGraphicsRule`:

```
\DeclareGraphicsRule{.eps.gz}{eps}{.eps.bb}{`gunzip -c #1}
```

Подчеркнём, что в аргументах *ext* и *read-file* перед названием расширения ставится точка.

Приведём теперь пример с декларацией `\DeclareGraphicsRule` по включению в документ фотографического изображения солнечного затмения из файла `s.jpg`. На странице 3 рассказано, как преобразовать рисунок из «чужого» для `dvips` формата JPEG в формат EPS. Итак, преобразуем сначала файл `s.jpg` в файл `s.eps`, затем создадим текстовый файл `s.jpg.bb` и скопируем в него из `eps`-файла строку с `BoundingBox`. Теперь файл `s.eps` можно удалить. Напечатаем во входном файле строку

```
\DeclareGraphicsRule{.jpg}{eps}{.jpg.bb}{`jpeg2ps -r 0 -h #1}
```

Теперь драйвер `dvips` знает, что ему надо делать с `jpg`-файлами: вызвать сначала программу `jpeg2ps`, которая преобразует рисунок из `jpg`-файла в формат EPS (без записи на диск, поскольку опция `-o` опущена), а затем вставить рисунок в документ. \LaTeX 'у на стадии вёрстки документа требуются параметры `BoundingBox`. Он возьмёт их из файла `s.jpg.bb`. После этого, команда

```
\includegraphics[width=3cm]{s.jpg}
```

во входном файле не вызовет ошибки ни \LaTeX 'а, ни `pdf \LaTeX` 'а:



```
\includegraphics[width=3cm]{s.jpg}
```

Вместо декларации `\DeclareGraphicsRule` можно воспользоваться ключами

```
ext=ext
type=type
read=read-file
command=command
```

команды `\includegraphics`. Ключи задают те же правила, что и одноимённые с ними аргументы декларации `\DeclareGraphicsRule`. Описанный выше пример с включением в документ рисунка из файла `s.jpg` при использовании ключей переписывается как


```
\includegraphics[width=3cm,%
  type=eps,read=.bb,command=`jpeg2ps -r 0 -h #1]{s.jpg}
```

или как

```
\includegraphics[width=3cm,%
  type=eps,ext=.jpg,read=.jpg.bb,command=`jpeg2ps -r 0 -h #1]{s}
```

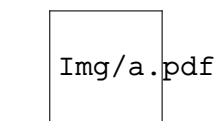
3.7 Черновой режим

На стадии подготовки документа можно использовать ключ

```
draft
```

который указывает, что вместо рисунка надо начертить рамку и напечатать внутри неё имя файла с рисунком. Пример:

```
\includegraphics[width=1.5cm,draft]{a}
```



Черновой режим обычно задаётся для всех рисунков сразу опцией `draft` либо в команде загрузки пакета, либо в `\documentclass`. В этом случае ключом

```
final
```

в команде `\includegraphics` можно отметить черновой режим для некоторых рисунков.

4 Манипуляции с боксами

Ключи команды `\includegraphics` позволяют манипулировать боксом, содержащим только сам рисунок. Пакет `graphicx` позволяет *любую* часть страницы документа оформить в виде бокса и манипулировать им, поворачивая, растягивая или сжимая его произвольным образом.

4.1 Изменение размеров бокса

4.1.1 Трансформация к указанному размеру

Команда

```
\resizebox{width}{height}{lr-text}
```

помещает, подобно команде `\mbox`, текст *lr-text* в бокс и затем сжимает или растягивает бокс вместе с содержимым так, чтобы его ширина и высота стали равны *width* и *height*:

```
\resizebox{4cm}{8mm}{abc}
```

abc

Сохранить отношение высоты к ширине бокса можно, указав в качестве *width* или *height* восклицательный знак `!`:

```
\resizebox{3cm}{!}{abc}
```

abc

Команда

```
\resizebox*{width}{totalheight}{lr-text}
```

действует подобно команде `\resizebox`, но подгоняет не высоту, а полную высоту бокса (сумма высоты и глубины) к указанному размеру *totalheight*.

На заметку В аргументах команды `\resizebox` исходные размеры бокса доступны в виде командных длин `\height`, `\width`, `\totalheight` и `\depth`.

4.1.2 Трансформация по указанному масштабу

Команда

```
\scalebox{h-scale}[v-scale]{lr-text}
```

помещает, подобно команде `\mbox`, текст *lr-text* в бокс и затем изменяет его ширину в *h-scale* раз и высоту в *v-scale* раз. Если опция *v-scale* опущена, то отношение ширины к высоте бокса при трансформации блока сохраняется. При отрицательных значениях аргументов происходит зеркальное отражение текста:

```
\scalebox{1.5}[3]{abc} \scalebox{1.5}%  
[-3]{abc}
```

4.2 Зеркальное отражение блока

Команда

```
\reflectbox{lr-text}
```

эквивалентна `\scalebox{-1}[1]{lr-text}`, описанной в разделе 4.1.2.

4.3 Поворот блока

Команда

```
\rotatebox[keyval-list]{angle}{lr-text}
```

помещает, подобно команде `\mbox`, текст *lr-text* в бокс и поворачивают его на *angle* градусов против часовой стрелки. По умолчанию бокс поворачивается относительно своей точки отсчёта⁵:

```
{\huge ghi \rotatebox{30}{ghi}}
```

Ключи *keyval-list* в необязательном аргументе команды `\rotatebox` позволяют изменить положение оси, вокруг которой поворачивается бокс, и единицу измерения угла поворота. Ключи в списке перечисляются через запятую и имеют вид *key=value*.

Ключ

```
origin=pos
```

позволяет указать одно из 12 предопределённых положений оси вращения, описанных в разделе 3.3.

Произвольное положение оси вращения можно задать с помощью ключей

```
x=dimen1  
y=dimen2
```

Например, если [*keyval-list*] имеет вид [*x*=4mm, *y*=3mm], то ось вращения проходит через точку, которая смещена относительно точки отсчёта бокса на 4 мм вдоль оси абсцисс и на 3 мм вдоль оси ординат.

⁵В этом и в следующем примере показаны базовая линия строки и границы боксов, занимаемых текстом до и после поворота.

В ключе

```
units=number
```

значение *number* соответствует полному обороту вокруг оси. Так, по умолчанию угол поворота измеряется в градусах, что соответствует `units=360`. Если задать `units=1`, то для поворота на 90 градусов надо в первом обязательном аргументе команды указать 0.25. Пример:

```
\rotatebox[origin=c,units=1]{0.25}{\huge W}
```



5 Альбомная ориентация страницы

Пакет `landscape` вводит окружение

```
\begin{landscape} ... \end{landscape}
```

содержание которого печатается на странице в альбомной ориентации. Колонтитулы и подстрочные примечания печатаются как на обычных страницах. При смене портретной ориентации страницы на альбомную и назад остаются не до конца заполненные страницы. Не работает внутри плавающих объектов.

6 Общие настройки графического пакета

В этом разделе приведены опции пакета `graphicx` и рассказано, как можно задавать значения ключей сразу для нескольких команд.

6.1 Опции пакета

`hiderotate` указывает, что не надо отображать в документе повернутые боксы.

`hidescale` указывает, что не надо отображать в документе боксы с изменённым размером.

`hiresbb` указывает, что надо читать параметры `%%HiResBoundingBox` вместо `%%BoundingBox` (см. описание ключа `hiresbb` на стр. 4).

`draft` | `final` включает (выключает) черновой режим вёрстки документа. В черновом режиме вместо рисунков рисуются рамки.

6.2 Установка ключей

Пакет `graphicx` загружает пакет `keyval`, который вводит декларацию

```
\setkeys{operation}{keyval-list}
```

позволяющую установить одни и те же значения ключей в опции *keyval-list* сразу нескольких команд `\rotatebox` (при значении *operation* равным `Grot`) и `\includegraphics` (при значении *operation* равным `Gin`). Так,

```
\setkeys{Gin}{width=\textwidth}
```

устанавливает ширину всех рисунков в области действия декларации `\setkeys`, равной ширине текста на странице.

Алфавитный указатель

d			
\DeclareGraphicsExtensions ...	7	преобразование в EPS	3
\DeclareGraphicsRule	8		
e		К	
EPS		Ключи	
BoundingBox	2	angle	6
HiResBoundingBox	4	bb	5
преобразование в PDF	3	clip	5
преобразование из JPEG	2	command	8
преобразование из TIFF	3	draft	9
		ext	8
g		final	9
\graphicspath	7	height	4
		hiresbb	4
i		keepaspectratio	5
\includegraphics	3	origin	6, 10
		read	8
j		scale	5
JPEG	3	totalheight	4
преобразование в EPS	2	trim	5
		type	8
l		units	11
landscape	11	viewport	5
		width	4
p		x	10
PDF		y	10
преобразование из EPS	3	глобальная установка	11
PNG	3		
r		О	
\reflectbox	10	Опции	
\resizebox	9	draft	11
\resizebox*	9	final	11
\rotatebox	10	hiderotate	11
		hidescale	11
s		hiresbb	11
\scalebox	10		
\setkeys	11	П	
		Пакеты	
t		keyval	11
TIFF	3	lscap	11