

Дополнительные пакеты (окончание)

Учебная практика

мехмат, III курс, кафедра ИВЭ

algorithm2e

Назначение

Параметры

Окружения

Метки

Слова

Блоки

Комментарии

Пример 1

Пример 2

if 1

if 2

Пример

switch

for

Стили

Русификация

ntheorem

siunitx 1

siunitx 2

Пакет `algorithm2e`

Назначение пакета algorithm2e

- ✓ Пакет предназначен для набора алгоритмов на псевдокоде.

Пример

```
\begin {algorithm}
  \KwIn {$a, b$}
  \KwResult {integer}
  \Begin { %
    \While {$a \neq b$,}
    { \eIf {$a > b$,}
      {$a \coloneqq
        a - b$\;}
      {$b \coloneqq
        b - a$\;}
    } %
    \Return $a$\;
  } %
\end {algorithm}
```

входные данные: a, b

результат: integer

начало

пока $a \neq b$, выполнять

если $a > b$, то

| $a := a - b$;

иначе

| $b := b - a$;

вернуть a ;

Параметры пакета

Таблица 1: основные параметры пакета `algorithm2e`

Параметр	Значение
<code>noline</code>	Блоки окружаются ключевыми словами «начало» и «конец», вертикальные черты не выводятся.
<code>lined</code>	То же, но между парами «начало» и «конец» выводятся вертикальные черты.
<code>vlined</code>	Выводится только «начало», в конце блока вертикальная черта имеет небольшое горизонтальное окончание: \lfloor
<code>linesnumbered</code>	В начале каждой строки выводится её номер.

[algorithm2e](#)

Назначение

Параметры

Окружения

Метки

Слова

Блоки

Комментарии

Пример 1

Пример 2

if 1

if 2

Пример

switch

for

Стили

Русификация

[ntheorem](#)

[siunitx 1](#)

[siunitx 2](#)

Окружения пакета

[algorithm2e](#)

[Назначение](#)

[Параметры](#)

[Окружения](#)

[Метки](#)

[Слова](#)

[Блоки](#)

[Комментарии](#)

[Пример 1](#)

[Пример 2](#)

[if 1](#)

[if 2](#)

[Пример](#)

[switch](#)

[for](#)

[Стили](#)

[Русификация](#)

[ntheorem](#)

[siunitx 1](#)

[siunitx 2](#)

Таблица 2: окружения пакета `algorithm2e`

Имя	Значение
<code>algorithm</code>	Основное окружение для определения алгоритмов.
<code>procedure</code>	Используется для кода процедуры. Заголовок (команда <code>\caption</code>) должен содержать пару круглых скобок.
<code>function</code>	Аналогично, но для функций.
<code>algorithm*</code> , <code>procedure*</code> , <code>function*</code>	Предназначены для формирования двухколоночных текстов.

Команды работы с метками

algorithm2e

Назначение

Параметры

Окружения

Метки

Слова

Блоки

Комментарии

Пример 1

Пример 2

if 1

if 2

Пример

switch

for

Стили

Русификация

ntheorem

siunitx 1

siunitx 2

`\nllabel` { \langle имя_метки \rangle }

Помечает заданную строку меткой при включённой автоматической нумерации. На метку можно, как обычно, сослаться при помощи команд `\ref` и т. д.

`\nl`

Отмечает заданную строку номером автоматически увеличивающегося счётчика.

`\lnl` { \langle имя_метки \rangle }

Отмечает заданную строку номером и помечает меткой.

`\lnlset` { \langle текст \rangle } { \langle имя_метки \rangle }

Отмечает заданную строку произвольным текстом и помечает меткой.

Команды основных ключевых слов

[algorithm2e](#)

[Назначение](#)

[Параметры](#)

[Окружения](#)

[Метки](#)

[Слова](#)

[Блоки](#)

[Комментарии](#)

[Пример 1](#)

[Пример 2](#)

[if 1](#)

[if 2](#)

[Пример](#)

[switch](#)

[for](#)

[Стили](#)

[Русификация](#)

[ntheorem](#)

[siunitx 1](#)

[siunitx 2](#)

\KwIn {⟨объявления⟩}

\KwOut {⟨объявления⟩}

\KwData {⟨объявления⟩}

\KwResult {⟨объявления⟩}

Печатает ⟨объявления⟩, предваряя их фразой «входные данные», «выходные данные» и т. д.

\Return {⟨значение⟩}

Печатает фразу «вернуть ⟨значение⟩».

\KwTo

Печатает слово «до».

Команды основных ключевых слов и блоков

algorithm2e

Назначение

Параметры

Окружения

Метки

Слова

Блоки

Комментарии

Пример 1

Пример 2

if 1

if 2

Пример

switch

for

Стили

Русификация

ntheorem

siunitx 1

siunitx 2

\;

Отмечает конец строки, печатает «;».

\Begin (*⟨команда_комментария⟩*) {*⟨фрагмент⟩*}

Выводит заданный фрагмент в оформлении блока. Если указана команда комментария в круглых скобках, она выводится на одной строке после слова «начало».

Команды комментариев

Таблица 3: команды комментариев

Команда	Выводит «;»	Выравнивание	Делает перевод строки
<code>\tcp {⟨Текст⟩}</code>		влево	✓
<code>\tcp* {⟨Текст⟩}</code>	;	вправо	✓
<code>\tcp* [r] {⟨Текст⟩}</code>	;	вправо	✓
<code>\tcp* [l] {⟨Текст⟩}</code>	;	влево	✓
<code>\tcp* [h] {⟨Текст⟩}</code>		влево	
<code>\tcp* [f] {⟨Текст⟩}</code>		вправо	
<code>\tcc</code> в тех же форматах — комментарии в стиле C			

[algorithm2e](#)

Назначение

Параметры

Окружения

Метки

Слова

Блоки

Комментарии

Пример 1

Пример 2

if 1

if 2

Пример

switch

for

Стили

Русификация

[ntheorem](#)

[siunitx 1](#)

[siunitx 2](#)

Пример комментариев

[algorithm2e](#)

[Назначение](#)

[Параметры](#)

[Окружения](#)

[Метки](#)

[Слова](#)

[Блоки](#)

[Комментарии](#)

Пример 1

[Пример 2](#)

[if 1](#)

[if 2](#)

[Пример](#)

[switch](#)

[for](#)

[Стили](#)

[Русификация](#)

[ntheorem](#)

[siunitx 1](#)

[siunitx 2](#)

Пример

```
\begin {algorithm}
  \nl (\tcp {tcp})\;
  \nl (\tcp* {tcp*})\;
  \nl (\tcp* [r] {tcp* [r]})\;
  \nl (\tcp* [l] {tcp* [l]})\;
  \nl (\tcp* [h] {tcp* [h]})\;
  \nl (\tcp* [f] {tcp* [f]})\;
\end {algorithm}
```

```
1 (// tcp
   );
2 (                                     // tcp*
   );
3 (                                     // tcp* [r]
   );
4 (; // tcp* [l]
   );
5 (// tcp* [h]);
6 (                                     // tcp* [f]);
```

Пример блока с комментарием

Пример

```
\begin {algorithm}
  \Begin (\tcp* [f] {begin})
  { %
    block 1\tcp* {block 1}
    \tcp {line}
    block 2\tcp* {block 2}
  } %
\end {algorithm}
```

```
начало // begin
┌ block 1; // block 1
│ // line
└ block 2; // block 2
```

[algorithm2e](#)

[Назначение](#)

[Параметры](#)

[Окружения](#)

[Метки](#)

[Слова](#)

[Блоки](#)

[Комментарии](#)

[Пример 1](#)

[Пример 2](#)

[if 1](#)

[if 2](#)

[Пример](#)

[switch](#)

[for](#)

[Стили](#)

[Русификация](#)

[ntheorem](#)

[siunitx 1](#)

[siunitx 2](#)

Команды условных операторов

[algorithm2e](#)

[Назначение](#)

[Параметры](#)

[Окружения](#)

[Метки](#)

[Слова](#)

[Блоки](#)

[Комментарии](#)

[Пример 1](#)

[Пример 2](#)

[if 1](#)

[if 2](#)

[Пример](#)

[switch](#)

[for](#)

[Стили](#)

[Русификация](#)

[ntheorem](#)

[siunitx 1](#)

[siunitx 2](#)

\If (*комментарий_then*) {*then*}

Выводит оператор «если–то».

\eIf (*комментарий_then*) {*then*} (*комментарий_else*)
{*else*}

Выводит оператор «если–то–иначе».

\Else (*комментарий_else*) {*else*}

Выводит часть «иначе». Необходима для завершения длинных условных операторов вида «если ... то ... иначе если ... ».

\ElseIf (*комментарий_else_if*) {*else_if*}

Выводит часть «иначе если».

Команды условных операторов (окончание)

Замечания:

- ✓ У всех перечисленных команд есть варианты «`\l...`», выводящие все части на одной строке.

Пример

```
\leIf {cond,} {a,} {b}
```

если *cond*, то *a*, иначе *b*;

- ✓ У всех команд, кроме `\eIf` есть варианты «`\u...`» без закрытия блока. Из таких частей можно составлять сложные условия.

[algorithm2e](#)

[Назначение](#)

[Параметры](#)

[Окружения](#)

[Метки](#)

[Слова](#)

[Блоки](#)

[Комментарии](#)

[Пример 1](#)

[Пример 2](#)

[if 1](#)

[if 2](#)

[Пример](#)

[switch](#)

[for](#)

[Стили](#)

[Русификация](#)

[ntheorem](#)

[siunitx 1](#)

[siunitx 2](#)

Пример сложного условного блока

Пример

```
\begin {algorithm}
  \uIf {$x < -1$,}
    {block 1\;}
  \uElseIf {$x > 1$,}
    {block 2\;}
  \Else
    {block 3\;}
\end {algorithm}
```

если $x < -1$, то
| block 1;
иначе если $x > 1$, то
| block 2;
иначе
└ block 3;

[algorithm2e](#)

[Назначение](#)

[Параметры](#)

[Окружения](#)

[Метки](#)

[Слова](#)

[Блоки](#)

[Комментарии](#)

[Пример 1](#)

[Пример 2](#)

[if 1](#)

[if 2](#)

[Пример](#)

[switch](#)

[for](#)

[Стили](#)

[Русификация](#)

[ntheorem](#)

[siunitx 1](#)

[siunitx 2](#)

Команды операторов выбора

algorithm2e

Назначение

Параметры

Окружения

Метки

Слова

Блоки

Комментарии

Пример 1

Пример 2

if 1

if 2

Пример

switch

for

Стили

Русификация

ntheorem

siunitx 1

siunitx 2

\Switch (*комментарий_switch*) {*условие*} {*блок*}

Выводит оператор выбора.

\Case (*комментарий*) {*условие*} {*ветвь*}

Выводит вариантную часть оператора выбора.

\Other (*комментарий*) {*ветвь*}

Выводит часть «иначе» оператора выбора.

Замечание: у команд `\Case` и `\Other` также есть варианты «`\l...`», у `\Case` дополнительно есть вариант «`\u...`». △

Команды операторов цикла

algorithm2e

Назначение

Параметры

Окружения

Метки

Слова

Блоки

Комментарии

Пример 1

Пример 2

if 1

if 2

Пример

switch

for

Стили

Русификация

ntheorem

siunitx 1

siunitx 2

\For (*комментарий*) {*условие*} {*блок*}

\ForEach (*комментарий*) {*условие*} {*блок*}

\ForAll (*комментарий*) {*условие*} {*блок*}

\While (*комментарий*) {*условие*} {*блок*}

\Repeat (*комментарий_repeat*) {*условие*} {*блок*}
(*комментарий_until*)

Замечание: у всех перечисленных команд есть варианты
«\l...».



Команды стилей

[algorithm2e](#)

[Назначение](#)

[Параметры](#)

[Окружения](#)

[Метки](#)

[Слова](#)

[Блоки](#)

[Комментарии](#)

[Пример 1](#)

[Пример 2](#)

[if 1](#)

[if 2](#)

[Пример](#)

[switch](#)

[for](#)

[Стили](#)

[Русификация](#)

[ntheorem](#)

[siunitx 1](#)

[siunitx 2](#)

`\FuncSty` {*⟨текст⟩*}

Выводит текст в стиле имени функции.

`\ArgSty` {*⟨текст⟩*}

Выводит текст в стиле аргумента функции.

`\SetKwFunction` {*⟨имя_команды⟩*} {*⟨отображение⟩*}

Определяет команду `\⟨имя_команды⟩ {⟨аргументы⟩}`, которая печатает «`⟨отображение⟩(⟨аргументы⟩)`» стилями имени и аргументов функции соответственно.

Пример

```
\SetKwFunction {myProc} {proc}
\myProc {args}
```

```
proc(args)
```

Русификация пакета

- ✓ В настоящее время пакет не поддерживает русского языка, однако в нём есть средства для определения ключевых слов, при помощи которых определены уже рассмотренные команды. С их помощью можно переопределить их для вывода других сообщений.

Пример

```
\SetKwBlock {Begin} {начало} {конец}

\SetKwIF %
  {If} {ElseIf} {Else} %
  {если} {то} {иначе если} {иначе} {конец}

\SetKwFor %
  {ForEach} {для каждого} {выполнять} {конец}
% ...
```

[algorithm2e](#)

[Назначение](#)

[Параметры](#)

[Окружения](#)

[Метки](#)

[Слова](#)

[Блоки](#)

[Комментарии](#)

[Пример 1](#)

[Пример 2](#)

[if 1](#)

[if 2](#)

[Пример](#)

[switch](#)

[for](#)

[Стили](#)

[Русификация](#)

[nththeorem](#)

[siunitx 1](#)

[siunitx 2](#)

algorithm2e

ntheorem

Назначение

Параметры

Определение

Команды 1

Стили 1

Стили 2

Стили 3

Команды 2

Команды 3

Пример 1

Пример 2

siunitx 1

siunitx 2

Пакет `ntheorem`

Назначение пакета `ntheorem`

[algorithm2e](#)

[ntheorem](#)

Назначение

[Параметры](#)

[Определение](#)

[Команды 1](#)

[Стили 1](#)

[Стили 2](#)

[Стили 3](#)

[Команды 2](#)

[Команды 3](#)

[Пример 1](#)

[Пример 2](#)

[siunitx 1](#)

[siunitx 2](#)

- ✓ Пакет предназначен для оформления окружений типа теорем (теоремы, леммы, доказательства, следствия, замечания и т. д.)
- ✓ Пакет дополняет стандартные возможности \LaTeX (команда `\newtheorem`) и позволяет в более широких пределах управлять внешним видом окружений.
- ✓ Также на основе механизмов AUX-файлов пакет реализует автоматическое размещение символов окончания окружений: «□» и т. п. (для построения окончательной версии документа может потребоваться нескольких проходов \LaTeX).

Основные параметры пакета

Таблица 4: основные параметры пакета `ntheorem`

Параметр	Значение
<code>amsmath</code>	Используется при совместном использовании пакета с пакетом <code>amsmath</code> . Пакет <code>amsmath</code> должен подключаться перед пакетом <code>ntheorem</code> .
<code>hyperref</code>	То же самое для пакета <code>hyperref</code> .
<code>framed</code>	Позволяет определять теоремы внутри рамок. По умолчанию теоремы отделяются от текста горизонтальными линиями, что позволяет делать внутри них переносы страниц.
<code>thmmarks</code>	Включает возможность автоматического размещения символов в конце теорем.

[algorithm2e](#)

[ntheorem](#)

Назначение

Параметры

Определение

Команды 1

Стили 1

Стили 2

Стили 3

Команды 2

Команды 3

Пример 1

Пример 2

[siunitx 1](#)

[siunitx 2](#)

Команда определения теоремы

[algorithm2e](#)

[ntheorem](#)

Назначение

Параметры

Определение

Команды 1

Стили 1

Стили 2

Стили 3

Команды 2

Команды 3

Пример 1

Пример 2

[siunitx 1](#)

[siunitx 2](#)

`\newtheorem` { \langle имя_окружения \rangle } [\langle имя_окружения₂ \rangle]
{ \langle заголовок \rangle }

Создаёт окружение с заданным именем, устанавливая для него заданный заголовок. Нумерация окружения будет общей с окружением \langle имя_окружения₂ \rangle , если оно указано. Команда аналогична стандартной за исключением того, что она учитывает ранее установленные настройки внешнего вида, определённые при помощи остальных команд пакета.

Команды определения стиля теоремы

algorithm2e

ntheorem

Назначение

Параметры

Определение

Команды 1

Стили 1

Стили 2

Стили 3

Команды 2

Команды 3

Пример 1

Пример 2

siunitx 1

siunitx 2

`\theoremstyle` { \langle ИМЯ_СТИЛЯ \rangle }

Определяет имя стиля, который будут использовать теоремы, определённые далее при помощи команды `\newtheorem` (табл. 5, 6, 7).

`\theoremheaderfont` { \langle КОМАНДЫ \rangle }

Определяет команды, влияющие на шрифт заголовка теоремы.

`\theorembodyfont` { \langle КОМАНДЫ \rangle }

Определяет команды, влияющие на шрифт текста теоремы.

Предопределённые стили теорем

Таблица 5: стили теорем, определённые в пакете

Стиль	Пример
plain	Теорема 1: <i>текст теоремы</i> <input type="checkbox"/>
break	Теорема 2: <i>текст теоремы</i> <input type="checkbox"/>
change	3 Теорема: <i>текст теоремы</i> <input type="checkbox"/>
changebreak	4 Теорема: <i>текст теоремы</i> <input type="checkbox"/>

[algorithm2e](#)

[ntheorem](#)

Назначение

Параметры

Определение

Команды 1

Стили 1

Стили 2

Стили 3

Команды 2

Команды 3

Пример 1

Пример 2

[siunitx 1](#)

[siunitx 2](#)

Предопределённые стили теорем (продолжение)

Таблица 6: стили теорем, определённые в пакете (продолжение)

Стиль	Пример
margin	5 Теорема: <i>текст теоремы</i> <input type="checkbox"/>
marginbreak	6 Теорема: <i>текст теоремы</i> <input type="checkbox"/>
nonumberplain	Теорема: <i>текст теоремы</i> <input type="checkbox"/>
nonumberbreak	Теорема: <i>текст теоремы</i> <input type="checkbox"/>

[algorithm2e](#)

[ntheorem](#)

Назначение

Параметры

Определение

Команды 1

Стили 1

Стили 2

Стили 3

Команды 2

Команды 3

Пример 1

Пример 2

[siunitx 1](#)

[siunitx 2](#)

Предопределённые стили теорем (окончание)

Таблица 7: стили теорем, определённые в пакете (окончание)

Стиль	Пример
empty	

[algorithm2e](#)

[ntheorem](#)

Назначение

Параметры

Определение

Команды 1

Стили 1

Стили 2

Стили 3

Команды 2

Команды 3

Пример 1

Пример 2

[siunitx 1](#)

[siunitx 2](#)

Команды определения стиля теоремы (продолжение)

algorithm2e

ntheorem

Назначение

Параметры

Определение

Команды 1

Стили 1

Стили 2

Стили 3

Команды 2

Команды 3

Пример 1

Пример 2

siunitx 1

siunitx 2

`\theoremnumbering` { \langle имя_стиля_нумерации \rangle }

Определяет имя стиля для нумерации теорем. Возможные значения: **arabic** (арабские цифры, по умолчанию), **Roman** (римские цифры), **Alph** (заглавные латинские буквы) и т. д.

`\theoremseparator` { \langle текст \rangle }

Определяет текст, отделяющий заголовок теоремы от её содержания («:» и т. д.)

`\theorempreskip` { \langle длина \rangle }

`\theorempostskip` { \langle длина \rangle }

Определяют вертикальное расстояние остального текста до/после теоремы.

Команды определения стиля теоремы (окончание)

algorithm2e

ntheorem

Назначение

Параметры

Определение

Команды 1

Стили 1

Стили 2

Стили 3

Команды 2

Команды 3

Пример 1

Пример 2

siunitx 1

siunitx 2

`\theoremindent` { \langle размер \rangle }

Определяет размер горизонтального отступа слева абзаца с теоремой.

`\theoremsymbol` { \langle текст \rangle }

Определяет текст, автоматически помещаемый справа внизу окружения теоремы. Учитывается только при указанном параметре пакета `thmmarks`.

Пример определения стилей

[algorithm2e](#)

[ntheorem](#)

Назначение

Параметры

Определение

Команды 1

Стили 1

Стили 2

Стили 3

Команды 2

Команды 3

Пример 1

Пример 2

[siunitx 1](#)

[siunitx 2](#)

```
\RequirePackage {ifpdf} %  
  
\ifpdf  
  %  
  \PassOptionsToPackage {hyperref} {ntheorem}  
  %  
\fi  
  
\usepackage [thmmarks, amsmath] {ntheorem}  
\usepackage {amssymb} % \Box
```

Пример определения стилей (окончание)

algorithm2e

ntheorem

Назначение

Параметры

Определение

Команды 1

Стили 1

Стили 2

Стили 3

Команды 2

Команды 3

Пример 1

Пример 2

siunitx 1

siunitx 2

```
\theorempreskipamount \medskipamount %
\theorempostskipamount \medskipamount %

\theoremstyle {nonumberplain} %
\theoremseparator {.} %
\theoremsymbol {\ensuremath {\_ \diamondsuit}} %
\newtheorem {definition} {Определение} %
\theoremstyle {nonumberbreak} %
\newtheorem {definitions} {Определения} %

% ...
```

algorithm2e

ntheorem

siunitx 1

Назначение

Пример 1

Команды 1

Команды 2

Команды 3

Команды 4

Пример 2

Команды 5

Команды 6

Колонки

Таблицы

siunitx 2

Пакет `siunitx` (введение)

Назначение пакета `siunitx`

[algorithm2e](#)

[ntheorem](#)

[siunitx 1](#)

Назначение

[Пример 1](#)

[Команды 1](#)

[Команды 2](#)

[Команды 3](#)

[Команды 4](#)

[Пример 2](#)

[Команды 5](#)

[Команды 6](#)

[Колонки](#)

[Таблицы](#)

[siunitx 2](#)

Пакет предназначен для использования в текстовом и математическом режиме. Его назначение состоит в автоматическом форматировании следующих элементов:

- ✓ целых чисел и чисел с фиксированной и плавающей запятой;
- ✓ списков и диапазонов чисел;
- ✓ физических величин, как входящих в систему СИ, так и не принадлежащих ей;
- ✓ комбинаций вышеперечисленных элементов;
- ✓ колонок таблиц с числами и единицами измерения.

Замечание: для локализации пакета необходимо установить его настройки (разделитель целой и дробной части, разделитель последнего числа в списке и т. д.), а также переопределить используемые единицы измерения.



Пример локализации пакета siunitx

[algorithm2e](#)

[nththeorem](#)

[siunitx 1](#)

Назначение

Пример 1

Команды 1

Команды 2

Команды 3

Команды 4

Пример 2

Команды 5

Команды 6

Колонки

Таблицы

[siunitx 2](#)

```
\usepackage {siunitx}

\sisetup %
{ %
  output-decimal-marker = {,}, % десятичная запятая
  list-final-separator = { и~}, % разделитель списка
  list-pair-separator = { и~}, % разделитель пары
  range-phrase = {~\ldots\ }, % разделитель диапазона
} %

\DeclareSIUnit \kilogram {\text {кг}} %
\DeclareSIUnit \metre {\text {м}} %
% И т. д.
```

Команды пакета siunitx

algorithm2e

ntheorem

siunitx 1

Назначение

Пример 1

Команды 1

Команды 2

Команды 3

Команды 4

Пример 2

Команды 5

Команды 6

Колонки

Таблицы

siunitx 2

\sisetup {*⟨настройки⟩*}

Глобально устанавливает настройки для остальных команд. Для каждой команды также можно задавать локальные настройки. Некоторые из основных настроек будут приведены далее.

\num [*⟨настройки⟩*] {*⟨число⟩*}

Выводит заданное число в отформатированном виде.

\ang [*⟨настройки⟩*] {*⟨угол⟩*}

Выводит заданное значение угла в градусах. *⟨угол⟩* может быть задан как в виде вещественного числа, так и в виде списка из (возможно, пустых) значений градусов, минут и секунд, разделённых символом «;».

Команды пакета siunitx (продолжение)

algorithm2e

ntheorem

siunitx 1

Назначение

Пример 1

Команды 1

Команды 2

Команды 3

Команды 4

Пример 2

Команды 5

Команды 6

Колонки

Таблицы

siunitx 2

\numlist [*⟨настройки⟩*] {⟨числа⟩}

Выводит заданные числа в виде списка («1, 2, 3 и 4»).
⟨числа⟩ должны разделяться символом «;». Команда
должна использоваться только в текстовом режиме.

\numrange [*⟨настройки⟩*] {⟨число₁⟩} {⟨число₂⟩}

Выводит заданные числа в виде диапазона («1 ... 4»).
Эта команда также должна использоваться в текстовом
режиме.

Команды пакета `siunitx` (продолжение)

algorithm2e

ntheorem

siunitx 1

Назначение

Пример 1

Команды 1

Команды 2

Команды 3

Команды 4

Пример 2

Команды 5

Команды 6

Колонки

Таблицы

siunitx 2

`\si` [*⟨настройки⟩*] {⟨единица⟩}

Выводит заданную единицу измерения в отформатированном виде. Вывод производится в математическом режиме. {⟨единица⟩} может быть задана как в кратком формате («`kg.m.s^{-1}`» — допустимы только английские буквы, так как вывод осуществляется в математическом режиме), так и в виде последовательности макросов. Макросы могут представлять как единицы измерения (`\kilogram`, `\ohm` и т. д.), так и операции над ними: `\per` («в»), `\square` («квадратный»), `\cubic` («кубический»), `\tothe {⟨число⟩}` («в степени...»).

Команды пакета siunitx (продолжение)

[algorithm2e](#)

[ntheorem](#)

[siunitx 1](#)

Назначение

Пример 1

Команды 1

Команды 2

Команды 3

Команды 4

Пример 2

Команды 5

Команды 6

Колонки

Таблицы

[siunitx 2](#)

\SI [*⟨настройки⟩*] {⟨число⟩} [*⟨префикс⟩*] {⟨единица⟩}

Является комбинацией команд `\num` и `\si`. *⟨префикс⟩* может задавать единицу, печатаемую перед числом (как правило, обозначение денежной единицы).

\SIlist [*⟨настройки⟩*] {⟨числа⟩} {⟨единица⟩}

Является комбинацией команд `\numlist` и `\si`.

По умолчанию *⟨единица⟩* печатается после каждого из значений списка (управляется настройкой `list-units`).

\SIrange [*⟨настройки⟩*] {⟨число₁⟩} {⟨число₂⟩} {⟨единица⟩}

Является комбинацией команд `\numrange` и `\si`.

По умолчанию *⟨единица⟩* печатается после каждого из обоих значений (управляется настройкой `range-units`).

Пример использования команд пакета siunitx

[algorithm2e](#)

[ntheorem](#)

[siunitx 1](#)

Назначение

Пример 1

Команды 1

Команды 2

Команды 3

Команды 4

Пример 2

Команды 5

Команды 6

Колонки

Таблицы

[siunitx 2](#)

Пример

<code>\num {\pm 12345.67891} \\</code>	$\pm 12\,345,678\,91$
<code>\num {9.7(3) + 2.1 i} \\</code>	$9,7(3) + 2,1i$
<code>\num {1.2 \times 3.1 / 5e2} \\</code>	$1,2 \times 3,1 / (5 \times 10^2)$
<code>\ang [color = blue] {42; 15; } \\</code>	$42^\circ 15'$
<code>\numlist {1.2e-3; e-2; 2.4} \\</code>	$1,2 \times 10^{-3}, 10^{-2}$ и $2,4$
<code>\numrange {2.1} {3.4} \\</code>	$2,1 \dots 3,4$
<code>\si {\kilogram \per \cubic \metre} \\</code>	кг/м^3
<code>\SI {4.55e-3} {\square \metre} \\</code>	$4,55 \times 10^{-3} \text{ м}^2$
<code>\SIlist {1;2;3} {\kilogram} \\</code>	$1 \text{ кг}, 2 \text{ кг}$ и 3 кг
<code>\SIrange {-3.1} {2.5} {\metre} \\</code>	$-3,1 \text{ м} \dots 2,5 \text{ м}$

Команды пакета siunitx (продолжение)

[algorithm2e](#)

[ntheorem](#)

[siunitx 1](#)

Назначение

Пример 1

Команды 1

Команды 2

Команды 3

Команды 4

Пример 2

Команды 5

Команды 6

Колонки

Таблицы

[siunitx 2](#)

`\DeclareSIUnit` $\langle \text{имя} \rangle \{ \langle \text{определение} \rangle \}$

Определяет новую единицу измерения с заданным именем. $\langle \text{определение} \rangle$ может содержать код \LaTeX для непосредственного вывода, имена других единиц измерения, степени и макрос \per . Может использоваться только в преамбуле.

Пример

```
\DeclareSIUnit \kgpmq {\kilogram \per \cubic \metre}
```

Пример

```
\SI {200} {\kgpmq}
```

200 кг/м³

Команды пакета siunitx (окончание)

[algorithm2e](#)

[ntheorem](#)

[siunitx 1](#)

Назначение

Пример 1

Команды 1

Команды 2

Команды 3

Команды 4

Пример 2

Команды 5

Команды 6

Колонки

Таблицы

[siunitx 2](#)

\DeclareSIPrefix \langle имя \rangle { \langle определение \rangle } { \langle степень_10 \rangle }

\DeclareBinaryPrefix \langle имя \rangle { \langle определение \rangle } { \langle степень_2 \rangle }

Определяют, соответственно, новые десятичные и двоичные префиксы для единиц измерения.

Пример

```
\DeclareSIPrefix {\kilo} {\text {к}} {3} %  
\DeclareBinaryPrefix {\kibi} {\text {Ки}} {10}
```

Определение колонок таблиц

[algorithm2e](#)

[ntheorem](#)

[siunitx 1](#)

Назначение

Пример 1

Команды 1

Команды 2

Команды 3

Команды 4

Пример 2

Команды 5

Команды 6

Колонки

Таблицы

[siunitx 2](#)

Пакет определяет дополнительные типы колонок таблиц:

s [*⟨настройки⟩*]

Определяет колонку из чисел, выровненных по умолчанию относительно десятичной запятой и других элементов (регулируется настройками).

s [*⟨настройки⟩*]

Определяет колонку из единиц измерения, выровненных по умолчанию по центру (регулируется настройками).

Печать чисел внутри таблиц

algorithm2e

ntheorem

siunitx 1

Назначение

Пример 1

Команды 1

Команды 2

Команды 3

Команды 4

Пример 2

Команды 5

Команды 6

Колонки

Таблицы

siunitx 2

\tablenum [*настройки*] {*число*}

Выводит число внутри ячейки таблицы. Может использоваться внутри команд `\multicolumn` и `\multirow` с сохранением выравнивания числа относительно других чисел текущей колонки.

Пример

```
\begin {tabular} {s [color = blue] S}
  \kilogram & 2.45 \\
  \metre    & 1.3  \\
  \multicolumn {2} {c} {\tablenum {1234.56}} \\
  \multicolumn {2} {c} {\tablenum {23.356}} \\
\end {tabular}
```

кг	2,45
м	1,3
	1234,56
	23,356

[algorithm2e](#)

[ntheorem](#)

[siunitx 1](#)

[siunitx 2](#)

Вид 1

Вид 2

Вид 3

Списки

Единицы

Формат 1

Формат 2

Формат 3

Округление 1

Округление 2

Округление 3

Шрифты 1

Шрифты 2

Пример 1

Таблицы 1

Таблицы 2

Таблицы 3

Таблицы 4

Пример 2

Пакет `siunitx` (настройки пакета)

Внешний вид единиц измерения и чисел

[algorithm2e](#)

[ntheorem](#)

[siunitx 1](#)

[siunitx 2](#)

Вид 1

[Вид 2](#)

[Вид 3](#)

[Списки](#)

[Единицы](#)

[Формат 1](#)

[Формат 2](#)

[Формат 3](#)

[Округление 1](#)

[Округление 2](#)

[Округление 3](#)

[Шрифты 1](#)

[Шрифты 2](#)

[Пример 1](#)

[Таблицы 1](#)

[Таблицы 2](#)

[Таблицы 3](#)

[Таблицы 4](#)

[Пример 2](#)

per-mode = reciprocal, ... (по умолчанию: **reciprocal**)

Определяет внешний вид единиц измерения, стоящих после макроса `\per`.

Таблица 8: значения ключа `per-mode`

Имя	Значение
reciprocal	Генерировать отрицательные степени.
symbol	Использовать символ «/».
fraction	Использовать команду <code>\frac</code> .

inter-unit-product = $\langle \text{код} \rangle$ (по умолчанию: `\,`)

Определяет код, разделяющий перемножаемые единицы.

Внешний вид чисел

[algorithm2e](#)

[ntheorem](#)

[siunitx 1](#)

[siunitx 2](#)

[Вид 1](#)

[Вид 2](#)

[Вид 3](#)

[Списки](#)

[Единицы](#)

[Формат 1](#)

[Формат 2](#)

[Формат 3](#)

[Округление 1](#)

[Округление 2](#)

[Округление 3](#)

[Шрифты 1](#)

[Шрифты 2](#)

[Пример 1](#)

[Таблицы 1](#)

[Таблицы 2](#)

[Таблицы 3](#)

[Таблицы 4](#)

[Пример 2](#)

exponent-base = $\langle \text{код} \rangle$ (по умолчанию: 10)

Определяет код, выводящий базовую часть экспоненты числа.

exponent-product = $\langle \text{код} \rangle$ (по умолчанию: `\times`)

Определяет код, отделяющий при выводе мантиссу от экспоненты числа.

output-decimal-marker = $\langle \text{код} \rangle$ (по умолчанию: `.`)

Определяет код, отделяющий при выводе целую часть от дробной.

group-separator = $\langle \text{код} \rangle$ (по умолчанию: `\,`)

Определяет код, отделяющий при выводе группы цифр.

Внешний вид чисел (окончание)

[algorithm2e](#)

[ntheorem](#)

[siunitx 1](#)

[siunitx 2](#)

[Вид 1](#)

[Вид 2](#)

[Вид 3](#)

[Списки](#)

[Единицы](#)

[Формат 1](#)

[Формат 2](#)

[Формат 3](#)

[Округление 1](#)

[Округление 2](#)

[Округление 3](#)

[Шрифты 1](#)

[Шрифты 2](#)

[Пример 1](#)

[Таблицы 1](#)

[Таблицы 2](#)

[Таблицы 3](#)

[Таблицы 4](#)

[Пример 2](#)

group-digits = true, false, ... (по умолчанию: **true**)

Определяет группировку цифр при выводе.

Таблица 9: значения ключа group-digits

Имя	Значение
false	Не группировать цифры.
true	Группировать цифры целой и дробной частей.
integer	Группировать цифры только целой части.
decimal	Группировать цифры только дробной части.

group-minimum-digits = $\langle \text{число} \rangle$ (по умолчанию: 5)

Определяет минимальное количество цифр, которые следует группировать.

Внешний вид списков и диапазонов

[algorithm2e](#)

[ntheorem](#)

[siunitx 1](#)

[siunitx 2](#)

[Вид 1](#)

[Вид 2](#)

[Вид 3](#)

Списки

[Единицы](#)

[Формат 1](#)

[Формат 2](#)

[Формат 3](#)

[Округление 1](#)

[Округление 2](#)

[Округление 3](#)

[Шрифты 1](#)

[Шрифты 2](#)

[Пример 1](#)

[Таблицы 1](#)

[Таблицы 2](#)

[Таблицы 3](#)

[Таблицы 4](#)

[Пример 2](#)

list-units = repeat, brackets, ... (по умолчанию: **repeat**)

range-units = repeat, brackets, ... (по умолчанию: **repeat**)

Определяют способ вывода единиц в списке и диапазоне.

Таблица 10: значения ключей `list-units` и `range-units`

Имя	Значение
repeat	Выводить единицы измерения после каждого числа.
single	Выводить единицы измерения только после последнего числа.
brackets	Окружить все числа скобками и вывести единицы измерения после них.

Дополнительные единицы измерения

[algorithm2e](#)

[ntheorem](#)

[siunitx 1](#)

[siunitx 2](#)

[Вид 1](#)

[Вид 2](#)

[Вид 3](#)

[Списки](#)

[Единицы](#)

[Формат 1](#)

[Формат 2](#)

[Формат 3](#)

[Округление 1](#)

[Округление 2](#)

[Округление 3](#)

[Шрифты 1](#)

[Шрифты 2](#)

[Пример 1](#)

[Таблицы 1](#)

[Таблицы 2](#)

[Таблицы 3](#)

[Таблицы 4](#)

[Пример 2](#)

binary-units = true, false (по умолчанию: **false**)

Загружает двоичные единицы измерения (`\bit`, `\byte`) вместе с соответствующими префиксами (`\kibi`, `\mebi` и т. д.) Эту настройку необходимо передавать пакету при загрузке.

Пример

```
\usepackage [binary-units] {siunitx}
```

Формат выводимых чисел

[algorithm2e](#)

[ntheorem](#)

[siunitx 1](#)

[siunitx 2](#)

[Вид 1](#)

[Вид 2](#)

[Вид 3](#)

[Списки](#)

[Единицы](#)

[Формат 1](#)

[Формат 2](#)

[Формат 3](#)

[Округление 1](#)

[Округление 2](#)

[Округление 3](#)

[Шрифты 1](#)

[Шрифты 2](#)

[Пример 1](#)

[Таблицы 1](#)

[Таблицы 2](#)

[Таблицы 3](#)

[Таблицы 4](#)

[Пример 2](#)

scientific-notation = true, ... (по умолчанию: **false**)

Определяет способ автоматического преобразования чисел к экспоненциальной форме.

Таблица 11: значения ключа `scientific-notation`

Имя	Значение
false	Отключение преобразования.
true	Порядок числа выбирается так, чтобы в целой части мантиссы была одна цифра.
fixed	Порядок числа выбирается из значения настройки fixed-exponent .
engineering	Порядок числа выбирается кратным 3.

Формат выводимых чисел (продолжение)

retain-explicit-plus = true, false (по умолчанию: **false**)

Включает режим, в котором явно указываемый унарный знак «+» перед числом не удаляется при выводе.

Пример

```
\num {+1} \\
```

1

```
\num [retain-explicit-plus] {+1}
```

+1

algorithm2e

ntheorem

siunitx 1

siunitx 2

Вид 1

Вид 2

Вид 3

Списки

Единицы

Формат 1

Формат 2

Формат 3

Округление 1

Округление 2

Округление 3

Шрифты 1

Шрифты 2

Пример 1

Таблицы 1

Таблицы 2

Таблицы 3

Таблицы 4

Пример 2

Формат выводимых чисел (окончание)

algorithm2e

ntheorem

siunitx 1

siunitx 2

Вид 1

Вид 2

Вид 3

Списки

Единицы

Формат 1

Формат 2

Формат 3

Округление 1

Округление 2

Округление 3

Шрифты 1

Шрифты 2

Пример 1

Таблицы 1

Таблицы 2

Таблицы 3

Таблицы 4

Пример 2

retain-unity-mantissa = true, false (по умолчанию: **true**)

Аналогично, определяет, нужно ли оставлять единичную мантиссу.

Пример

```
\num {1e4} \\
```

1×10^4

```
\num [retain-unity-mantissa = false] {1e4}
```

10^4

retain-zero-exponent = true, false (по умолчанию: **false**)

Аналогично, определяет, нужно ли оставлять и экспоненту с нулевым порядком.

Округление выводимых чисел

[algorithm2e](#)

[ntheorem](#)

[siunitx 1](#)

[siunitx 2](#)

[Вид 1](#)

[Вид 2](#)

[Вид 3](#)

[Списки](#)

[Единицы](#)

[Формат 1](#)

[Формат 2](#)

[Формат 3](#)

Округление 1

[Округление 2](#)

[Округление 3](#)

[Шрифты 1](#)

[Шрифты 2](#)

[Пример 1](#)

[Таблицы 1](#)

[Таблицы 2](#)

[Таблицы 3](#)

[Таблицы 4](#)

[Пример 2](#)

round-mode = off, figures, places (по умолчанию: **off**)

Определяет способ автоматического округления чисел.

Таблица 12: значения ключа round-mode

Имя	Значение
off	Отключение округления.
figures	Выполняется округление дробной части значимых цифр мантииссы к количеству, определяемому значением настройки round-precision .
places	Выполняется округление цифр дробной части всего числа к количеству, определяемому значением настройки round-precision .

Округление выводимых чисел (продолжение)

round-integer-to-decimal = true, false (по ум.: false)

Определяет, следует ли добавлять нули к дробной части целого числа в процессе округления.

Пример

<pre>\sisetup {round-mode = places, round-precision = 3}</pre>	
<pre>\num {1} \\</pre>	1
<pre>\num [round-integer-to-decimal] {1}</pre>	1,000

algorithm2e

ntheorem

siunitx 1

siunitx 2

Вид 1

Вид 2

Вид 3

Списки

Единицы

Формат 1

Формат 2

Формат 3

Округление 1

Округление 2

Округление 3

Шрифты 1

Шрифты 2

Пример 1

Таблицы 1

Таблицы 2

Таблицы 3

Таблицы 4

Пример 2

Округление выводимых чисел (окончание)

- algorithm2e
- ntheorem
- siunitx 1
- siunitx 2
- Вид 1
- Вид 2
- Вид 3
- Списки
- Единицы
- Формат 1
- Формат 2
- Формат 3
- Округление 1
- Округление 2
- Округление 3**
- Шрифты 1
- Шрифты 2
- Пример 1
- Таблицы 1
- Таблицы 2
- Таблицы 3
- Таблицы 4
- Пример 2

zero-decimal-to-integer = true, false (по ум.: false)

Определяет, следует ли отбрасывать при выводе нулевую дробную часть числа.

Пример

<code>\num {1.000} \</code>	1,000
<code>\num [zero-decimal-to-integer] {1.000}</code>	1

Замечание: при передаче пакету siunitx функций вычисления (округление, преобразование к экспоненциальной форме) следует учитывать, что пакет выполняет вычисления с 32-битной точностью. То есть при использовании вышеупомянутых функций необходимо следить за тем, чтобы целая, дробная части и порядок числа находились в диапазоне $-2\,147\,483\,648 \dots 2\,147\,483\,647$.



Определение шрифтов в окружающем тексте

algorithm2e

ntheorem

siunitx 1

siunitx 2

Вид 1

Вид 2

Вид 3

Списки

Единицы

Формат 1

Формат 2

Формат 3

Округление 1

Округление 2

Округление 3

Шрифты 1

Шрифты 2

Пример 1

Таблицы 1

Таблицы 2

Таблицы 3

Таблицы 4

Пример 2

detect-weight = true, false (по умолчанию: **false**)

detect-family = true, false (по умолчанию: **false**)

detect-shape = true, false (по умолчанию: **false**)

detect-mode = true, false (по умолчанию: **false**)

Определяют, следует ли выбирать, соответственно, жирность, семейство, форму шрифта и режим (текстовый/математический) для вывода чисел в соответствии с параметрами окружающего их текста.

Определение шрифтов в окружающем тексте (окончание)

algorithm2e

ntheorem

siunitx 1

siunitx 2

Вид 1

Вид 2

Вид 3

Списки

Единицы

Формат 1

Формат 2

Формат 3

Округление 1

Округление 2

Округление 3

Шрифты 1

Шрифты 2

Пример 1

Таблицы 1

Таблицы 2

Таблицы 3

Таблицы 4

Пример 2

detect-all

(по умолчанию: $\langle \text{нет} \rangle$)

detect-none

(по умолчанию: $\langle \text{нет} \rangle$)

Устанавливают, соответственно, значение **true** или **false** для четырёх предыдущих настроек.

detect-display-math = true, false (по умолчанию: **false**)

Определяет, следует ли выбирать шрифт в математических формулах на отдельных строках (окружения `displaymath` и т. д.) в соответствии с выбранным шрифтом внутри формулы (**true**) или окружающего её текста (**false**).

Пример определения шрифтов в соответствии с окружением

Пример

```
\sisetup {per-mode = fraction, detect-display-math}
\begin {tabular} {p{.4 \linewidth} | p{.4 \linewidth}}
  \textbf {Value: \SI {100} {\metre \per \second}} &
  \textbf {Value: \SI [detect-all] {100} {\metre \per \second}} \\
  \boldmath \[ v = \SI {100} {\metre \per \second} \] &
  \boldmath \[ v = \SI [detect-all] {100} {\metre \per \second} \] \\
\end {tabular}
```

Value: 100 $\frac{\text{M}}{\text{c}}$

$$v = 100 \frac{\text{M}}{\text{c}}$$

Value: 100 $\frac{\text{M}}{\text{c}}$

$$v = 100 \frac{\text{M}}{\text{c}}$$

algorithm2e

ntheorem

siunitx 1

siunitx 2

Вид 1

Вид 2

Вид 3

Списки

Единицы

Формат 1

Формат 2

Формат 3

Округление 1

Округление 2

Округление 3

Шрифты 1

Шрифты 2

Пример 1

Таблицы 1

Таблицы 2

Таблицы 3

Таблицы 4

Пример 2

Обработка таблиц

[algorithm2e](#)

[ntheorem](#)

[siunitx 1](#)

[siunitx 2](#)

[Вид 1](#)

[Вид 2](#)

[Вид 3](#)

[Списки](#)

[Единицы](#)

[Формат 1](#)

[Формат 2](#)

[Формат 3](#)

[Округление 1](#)

[Округление 2](#)

[Округление 3](#)

[Шрифты 1](#)

[Шрифты 2](#)

[Пример 1](#)

[Таблицы 1](#)

[Таблицы 2](#)

[Таблицы 3](#)

[Таблицы 4](#)

[Пример 2](#)

table-number-alignment = ... (по ум.: **center-decimal-marker**)

Определяет способ выравнивания чисел в колонке типа S таблицы. Значения не по умолчанию требуют явного резервирования места при помощи других настроек.

Таблица 13: значения ключа table-number-alignment

Имя	Значение
center-decimal-marker	Помещать десятичный разделитель в центре колонки.
left	Выравнивать по левому краю.
right	Выравнивать по правому краю.
center	Выравнивать по центру.

Обработка таблиц (продолжение)

[algorithm2e](#)

[ntheorem](#)

[siunitx 1](#)

[siunitx 2](#)

[Вид 1](#)

[Вид 2](#)

[Вид 3](#)

[Списки](#)

[Единицы](#)

[Формат 1](#)

[Формат 2](#)

[Формат 3](#)

[Округление 1](#)

[Округление 2](#)

[Округление 3](#)

[Шрифты 1](#)

[Шрифты 2](#)

[Пример 1](#)

[Таблицы 1](#)

[Таблицы 2](#)

[Таблицы 3](#)

[Таблицы 4](#)

[Пример 2](#)

table-unit-alignment = left, ... (по умолчанию: **center**)

table-text-alignment = left, ... (по умолчанию: **center**)

Определяют выравнивание, соответственно, единиц измерения в колонках типа *s* и текста (ограниченного фигурными скобками) в колонках *s* и *S* таблицы.

table-alignment = center, left, right (по умолчанию: *нет*)

Устанавливает одно значение для трёх предыдущих настроек одновременно.

Обработка таблиц (продолжение)

[algorithm2e](#)

[ntheorem](#)

[siunitx 1](#)

[siunitx 2](#)

[Вид 1](#)

[Вид 2](#)

[Вид 3](#)

[Списки](#)

[Единицы](#)

[Формат 1](#)

[Формат 2](#)

[Формат 3](#)

[Округление 1](#)

[Округление 2](#)

[Округление 3](#)

[Шрифты 1](#)

[Шрифты 2](#)

[Пример 1](#)

[Таблицы 1](#)

[Таблицы 2](#)

[Таблицы 3](#)

[Таблицы 4](#)

[Пример 2](#)

table-figures-integer = $\langle \text{число} \rangle$ (по умолчанию: 3)

table-figures-decimal = $\langle \text{число} \rangle$ (по умолчанию: 2)

table-figures-exponent = $\langle \text{число} \rangle$ (по умолчанию: 0)

table-figures-uncertainty = $\langle \text{число} \rangle$ (по умолчанию: 0)

Резервируют место под указываемое количество разрядов в, соответственно, целой, дробной части числа, порядке и периоде (для цифр в скобках) при выводе числа в колонке типа S.

Обработка таблиц (окончание)

[algorithm2e](#)

[ntheorem](#)

[siunitx 1](#)

[siunitx 2](#)

[Вид 1](#)

[Вид 2](#)

[Вид 3](#)

[Списки](#)

[Единицы](#)

[Формат 1](#)

[Формат 2](#)

[Формат 3](#)

[Округление 1](#)

[Округление 2](#)

[Округление 3](#)

[Шрифты 1](#)

[Шрифты 2](#)

[Пример 1](#)

[Таблицы 1](#)

[Таблицы 2](#)

[Таблицы 3](#)

[Таблицы 4](#)

[Пример 2](#)

table-sign-mantissa = true, false (по умолчанию: **false**)

table-sign-exponent = true, false (по умолчанию: **false**)

Аналогично, резервируют место под знак мантиссы и порядка.

table-format = *⟨формат⟩* (по умолчанию: *⟨нет⟩*)

Является удобной заменой шести предыдущим настройкам: позволяет указывать формат выводимого числа в краткой форме.

Пример установки формата в таблице

Пример

```
\sisetup {table-format = +2.4e+2}  
\setlength {\tabcolsep} {12pt}  
  
\begin {tabular} %  
  { | S [table-number-alignment = left] %  
    | S [table-number-alignment = right] | }  
  {Some list of values 1} & {Some list of values 2} \\  
  1.2345e3 & 1.2345e3 \\  
  -12.4e-12 & -12.4e-12 \\  
\end {tabular}
```

Some list of values 1	Some list of values 2
$1,2345 \times 10^3$	$1,2345 \times 10^3$
$-12,4 \times 10^{-12}$	$-12,4 \times 10^{-12}$

algorithm2e

ntheorem

siunitx 1

siunitx 2

Вид 1

Вид 2

Вид 3

Списки

Единицы

Формат 1

Формат 2

Формат 3

Округление 1

Округление 2

Округление 3

Шрифты 1

Шрифты 2

Пример 1

Таблицы 1

Таблицы 2

Таблицы 3

Таблицы 4

Пример 2