

# Математические формулы в LaTeX

Материал из Викиучебника — открытых книг для открытого мира

Одним из главных мотивов для Дональда Кнута, когда он начал разрабатывать исходную систему TeX, было создание чего-то, что позволяло бы просто записывать математические формулы, но при этом выглядело бы профессионально на этапе печати. Тот факт, что ему это удалось, скорее всего и был причиной того, что TeX (а позже и LaTeX) стал настолько популярным в научном сообществе. Возможность набора математических формул — одна из самых сильных сторон LaTeX. Но при этом, это очень объёмная тема из-за существования большого количества математических обозначений.

Если для вашего документа требуется всего несколько простых математических формул, обычный LaTeX предоставит вам большинство инструментов, которые вам смогут понадобиться. Если же вы пишете научную статью, содержащую множество сложных формул, пакет `amsmath` приносит некоторое количество новых команд, которые являются более мощными и гибкими, чем те, которые предоставляются базовым LaTeX. Пакет `mathtools` исправляет некоторые причуды `amsmath` и добавляет полезные настройки, символы и окружения в `amsmath`. Чтобы использовать любой из данных пакетов, включите в преамбулу создаваемого документа:

```
\usepackage{amsmath}
```

или

```
\usepackage{mathtools}
```

Пакет `mathtools` загружает пакет `amsmath` и, следовательно, нет необходимости указывать `\usepackage{amsmath}` в преамбуле, если используется `mathtools`.

## Содержание

### Математическое окружение

[Вставка "выключенных" математических формул внутри блоков текста](#)

### Математические символы

#### Греческие буквы

### Математические операторы

### Степени и индексы

### Дроби и биномы

[Непрерывные дроби](#)

[Умножение двух чисел](#)

### Корни

### Ряды и интегралы

### Скобки, фигурные скобки и разделители

[Автоматическое определение размеров](#)

[Ручное определение размеров](#)

### Matrices and arrays

[Matrices in running text](#)

### Adding text to equations

[Formatted text](#)

### Formatting mathematics symbols

[Accents](#)

### Color

### Plus and minus signs

### Controlling horizontal spacing

### Manually Specifying Formula Style

### Advanced Mathematics: AMS Math package

[Introducing dots in formulas](#)

[Dots](#)

[Write an equation with the align environment](#)

### List of mathematical symbols

### Summary

### Notes

[Further reading](#)

[External links](#)

## Математическое окружение

Системе LaTeX необходимо сообщить, когда текст, который вы вводите, является математической формулой. Это необходимо из-за того, что LaTeX набирает математическую нотацию иначе, чем обычный текст. Поэтому для данной цели объявлены специальные окружения. Их можно разделить на две категории в зависимости от того, как они представлены:

- text* - текст формулы отображается прямо в строке, внутри текста, где он объявлен. Например, я могу написать формулу  $a + a = 2a$  прямо в этом предложении.
- displayed* - для отображения формулы в отдельной строке.

Поскольку математические формулы требуют особых окружений, естественно, есть их соответствующие названия, которые вы можете использовать стандартным способом. Однако, в отличие от большинства других окружений, есть удобные сокращения для объявления ваших формул. Следующая таблица объединяет информацию о них:

Тип	Встроенная (в текст) формула	Выключенная (на отдельной строке) формула	Выключенная формула с автонумерацией
Окружение	math	displaymath	equation
Сокращение LaTeX	<code>\(...\)</code>	<code>\[...\]</code>	
Сокращение TeX	<code>\$. . . \$</code>	<code>\$\$ . . . \$\$</code>	
Комментарий			equation* (версия со звездочкой) убирает нумерацию, но требует использования amsmath

**Внимание:** Следует избегать использования `$$ . . . $$`, так как это может вызывать проблемы, особенно с макросами AMS-LaTeX. Кроме того, в случае возникновения проблем сообщения об ошибках могут оказаться бесполезными.

Окружения `equation*` и `displaymath` являются функционально эквивалентными.

Если вы набираете текст в обычном режиме, говорят, что вы находитесь в *текстовом режиме*, но когда вы печатаете в одной из этих математических сред, вы, как говорят, находитесь в *математическом режиме*, который имеет некоторые отличия от *текстового режима*:

- Большинство пробелов и разрывов строк не имеют никакого значения, поскольку все пробелы либо получены логически из математических выражений, либо должны быть указаны с помощью специальных команд, таких как `\quad`
- Пустые строки не допускаются. Только один абзац на формулу.
- Каждая буква считается именем переменной и будет набрана как таковая. Если вы хотите набрать обычный текст в формуле (обычный вертикальный шрифт и нормальный интервал), вам необходимо ввести текст с помощью специальных команд.

### Вставка "выключенных" математических формул внутри блоков текста

Чтобы некоторые операторы, такие как `\lim` или `\sum`, правильно отображались в некоторых математических окружениях (имеется ввиду `$. . . . . $`), может быть удобно написать класс `\displaystyle` внутри окружения. Это может увеличить длину строки, но приведет к более правильному отображению показателей и индексов для некоторых математических операторов. Например, `$\sum$` напечатает маленькую  $\sum$ , а `\displaystyle \sum` напечатает большую  $\sum$ , как в уравнениях (Работает только при включенном пакете AMSMATH). Можно принудительно настроить такое поведение для всех математических сред, объявив `\everymath{\displaystyle}` в самом начале (т.е. до `\begin{document}`).

## Математические символы

В математике существует достаточно много различных символов! Ниже приведены те, к которым можно получить доступ прямо с клавиатуры:

`+ - = ! / ( ) [ ] < > | ' : *`

Помимо тех, что перечислены выше, для ввода некоторых символов могут потребоваться отдельные команды. Это требуется, например, для ввода греческих букв, символов множества и отношений, стрелок, бинарных операторов, и т.д.

Для примера:

`\forall x \in X, \quad \exists y \leq \epsilon`

$\forall x \in X, \quad \exists y \leq \epsilon$

К счастью, есть инструмент, который может значительно упростить поиск команды для определенного символа. Найдите "Detexify" в разделе [external links](#) ниже. Другой вариант – посмотреть "Полный список символов LaTeX" в разделе [external links](#) ниже.

## Греческие буквы

Греческие буквы довольно часто используются в математике, но их достаточно просто набирать в *математическом режиме*. Вам просто нужно ввести название буквы после обратной косой черты: если первая буква названия строчная, вы получите строчную греческую букву, если первая буква названия заглавная (только первая буква), тогда вы получите прописную букву. Обратите внимание, что некоторые заглавные греческие буквы выглядят как латинские, поэтому они не предоставляются LaTeX (например, заглавные буквы *Alpha* и *Beta* это просто латинские "A" и "B" соответственно). Строчные буквы эпсилон, тета, каппа, фи, пи, ро и сигма представлены в двух разных версиях. Альтернативная (*variant* сокр. *var*) версия, создается добавлением "var" перед названием буквы:

```
\alpha, \Alpha, \beta, \Beta, \gamma, \Gamma, \pi, \Pi, \phi, \varphi, \mu, \Phi, \varPhi
```

 $\alpha, A, \beta, B, \gamma, \Gamma, \pi, \Pi, \phi, \varphi, \mu, \Phi, \varPhi$ 

Прокрутите вниз до [#List of mathematical symbols](#) чтобы увидеть полный список греческих символов.

## Математические операторы

Оператор – это функция, которая записывается с помощью слова: например, тригонометрические функции (sin, cos, tan), логарифмы и экспоненты (log, exp), пределы (lim), а также след и определитель (tr, det). В LaTeX многие из них определены как команды:

```
\cos(2\theta) = \cos^2 \theta - \sin^2 \theta
```

 $\cos(2\theta) = \cos^2 \theta - \sin^2 \theta$ 

Для некоторых операторов, таких как [Предел](#), нижний индекс помещается под оператором:

```
\lim\limits_{x \to \infty} \exp(-x) = 0
```

 $\lim_{x \rightarrow \infty} \exp(-x) = 0$ 

Для оператора [Сравнения по модулю](#) существует две команды: `\bmod` и `\pmod`:

```
a \bmod b
```

 $a \bmod b$ 

```
x \equiv a \pmod{b}
```

 $x \equiv a \pmod{b}$ 

Чтобы использовать операторы, которые не определены заранее, например `argmax`, см. [\[../Высшая математика#0определяемые операторы|определяемые операторы\]](#)

## Степени и индексы

Степени и индексы эквивалентны верхним и нижним индексам в обычном текстовом режиме. Символ каретки (^; так же известный как [циркумфлекс](#)) используется чтобы что-то поднять, а нижнее подчеркивание (\_) для опускания. Если необходимо повысить или понизить выражение, содержащее больше одного символа, его необходимо сгруппировать с помощью фигурных скобок ({ и }).

```
k_{n+1} = n^2 + k_n^2 - k_{n-1}
```

 $k_{n+1} = n^2 + k_n^2 - k_{n-1}$ 

Для степеней, состоящих из более чем одной цифры, заключите степень в {}.

```
n^{22}
```

 $n^{22}$

Подчёркивание (`\_`) может использоваться с вертикальной чертой (`|`) при использовании выражения в качестве нижнего индекса (*это предложение требует уточнения*):

```
f(n) = n^5 + 4n^2 + 2 |_{n=17}
```

$$f(n) = n^5 + 4n^2 + 2|_{n=17}$$

## Дроби и биномы

Дроби создаются с помощью команды `\frac{numerator}{denominator}`. Так же и Биномиальный коэффициент можно записать используя команду `\binom`:

```
\frac{n!}{k!(n-k)!} = \binom{n}{k}
```

$$\frac{n!}{k!(n-k)!} = \binom{n}{k}$$

Дроби можно помещать одну внутри другой:

```
\frac{\frac{1}{x} + \frac{1}{y}}{y-z}
```

$$\frac{\frac{1}{x} + \frac{1}{y}}{y-z}$$

Так же обратите внимание, что при встраивании дробей в строку текста или записи одной дроби внутри другой,  $\frac{a}{b}$ , их отображаемый размер должен быть заметно меньше, чем в выключенной формуле. Команды `\tfrac` и `\dfrac` позволяют использовать стиль записи соответствующий использованию `\textstyle` и `\displaystyle`. Точно так же работают команды `\tbinom` и `\dbinom`, записывающие биномиальный коэффициент.

Для относительно простых дробей, особенно внутри текста, может быть более эстетично использовать Степени и индексы:

```
^3/_7
```

$$^3/_7$$

Данная запись может показаться немного "растянутой" (занимающей много места), сжатую версию можно определить, вставив некоторое отрицательное пространство.

```
%running fraction with slash - requires math mode.
\newcommand*\rfrac[2]{{}^{\scriptstyle\#1}/_{\scriptstyle\#2}}
\rfrac{3}{7}
```

$$^3/_7$$

Если вам необходимо часто использовать подобную запись дробей в своём документе, рекомендуем использовать пакет *xfrac*. Данный пакет поддерживает команду `\sfrac` для создания наклонных дробей. Использование:

```
Take $\sfrac{1}{2}$ cup of sugar, \dots
3\times\sfrac{1}{2}=1\sfrac{1}{2}

Take ${}^{1/2}$ cup of sugar, \dots
3\times{}^{1/2}=1^{1/2}
```

Take  $\frac{1}{2}$  cup of sugar, ...

$$3 \times \frac{1}{2} = 1\frac{1}{2}$$

Take  $\frac{1}{2}$  cup of sugar, ...

$$3 \times \frac{1}{2} = 1\frac{1}{2}$$

Если в качестве показателя степени используются дроби, необходимо использовать фигурные скобки вокруг команды `\sfrac`:

```
$x^{\sfrac{1}{2}}$ % no error
$x^{\sfrac{1}{2}}$ % error
$x^{\sfrac{1}{2}}$ % no error
```

```
$x^{\frac{1}{2}}$ % no error
```

$$x^{\frac{1}{2}}$$

В некоторых случаях добавление только данного пакета может привести к ошибкам о том, что определённые формы шрифтов недоступны. Тогда необходимо так же добавить пакеты *lmodern* и *fix-cm*.

В качестве альтернативы, пакет `\int\limits_a^b\frac{1}{(1+x)^{-3/2}}\mathrm{d}x= \left.-\frac{1}{\sqrt{1+x}}\right|_a^b` предоставляет команду **`\nicefrac`** использование которой аналогично использованию **`\sfrac`**.

Непрерывные дроби

Непрерывные дроби следует записывать с помощью команды **`\cfrac`**:

```
\begin{equation}
x = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cfrac{1}{a_4}}}}
\end{equation}
```

$$x = a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{a_3 + \frac{1}{a_4}}}}$$

Умножение двух чисел

Чтобы сделать умножение визуально похожим на дробь, можно использовать вложенный массив, например, умножение чисел, написанных одно под другим..

```
\begin{equation}
\frac{
\begin{array}{b}
\left( x_1 x_2 \right) \\
\times \left( x'_1 x'_2 \right) \\
\end{array}
}{
\left( y_1 y_2 y_3 y_4 \right)
}
\end{equation}
```

$$\frac{\begin{array}{b} (x_1 x_2) \\ \times (x'_1 x'_2) \end{array}}{(y_1 y_2 y_3 y_4)}$$

Корни

Команда **`\sqrt`** создаёт символ квадратного корня, окружающий математическое выражение. Он принимает необязательный аргумент в квадратных скобках ( [ и ] ) для изменения показателя (степени) корня:

```
\sqrt{\frac{a}{b}}
```

$$\sqrt{\frac{a}{b}}$$

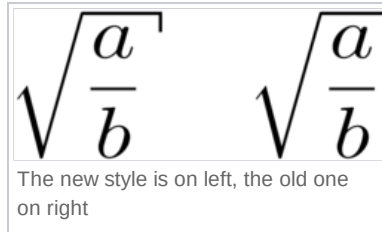
```
\sqrt[n]{1+x+x^2+x^3+\dots+x^n}
```

$$\sqrt[n]{1+x+x^2+x^3+\dots+x^n}$$

Some people prefer writing the square root "closing" it over its content. This method arguably makes it more clear what is in the scope of the root sign. This habit is not normally used while writing with the computer, but if you still want to change the output of the square root, LaTeX gives you this possibility. Just add the following code in the preamble of your document:

```
% New definition of square root:
% it renames \sqrt as \oldsqrt
\let\oldsqrt\sqrt
% it defines the new \sqrt in terms of the old one
\def\sqrt{\mathpalette\DHLhksqrt}
\def\DHLhksqrt#1#2{%
\setbox0=\hbox{$#1\oldsqrt{#2}$}\dimen0=\ht0
\advance\dimen0-0.2\ht0
```

```
\setbox2=\hbox{\vrule height\ht0 depth -\dimen0}%
{\box0\lower0.4pt\box2}}
```



This TeX code first renames the `\sqrt` command as `\oldsqrt`, then redefines `\sqrt` in terms of the old one, adding something more. The new square root can be seen in the picture on the left, compared to the old one on the right. Unfortunately this code won't work if you want to use multiple roots: if you try to write  $\sqrt[n]{a}$  as `\sqrt[n]{a}` after you used the code above, you'll just get a wrong output. In other words, you can redefine the square root this way only if you are not going to use multiple roots in the whole document.

An alternative piece of TeX code that does allow multiple roots is

```
\usepackage{letltxmacro}
\makeatletter
\let\oldr@@t\r@@t
\def\r@@t#1#2{%
\setbox0=\hbox{$\oldr@@t#1\{#2\},}$}\dimen0=\ht0
\advance\dimen0-0.2\ht0
\setbox2=\hbox{\vrule height\ht0 depth -\dimen0}%
{\box0\lower0.4pt\box2}}
\let\LTxMacro\oldsqrt
\renewcommand*\sqrt[2][\ ]{\oldsqrt[#1]{#2}}
\makeatother

$\sqrt{a}{b} \quad \oldsqrt{a}{b}$
```



However this requires the `\usepackage{letltxmacro}` package

## Ряды и интегралы

Команды `\sum` и `\int` создают символы Ряды и Интеграла соответственно. Нижний предел задаётся символом "\_", а верхний "^". Пример использования Ряды:

```
\sum_{i=1}^{10} t_i
```

$$\sum_{i=1}^{10} t_i$$

или

```
\displaystyle\sum_{i=1}^{10} t_i
```

$$\sum_{i=1}^{10} t_i$$

Пределы для интегралов задаются по правилам такой же нотации, что и для Рядов. Так же важно обозначать дифференциал некоторой величины прямым шрифтом, что достигается командой `\mathrm`, и с небольшим отступом от подынтегрального выражения с помощью команды `\,`,

```
\int_0^{\infty} e^{-x}\,\mathrm{d}x
```

$$\int_0^{\infty} e^{-x} \, dx$$

Есть много других "больших" команд, которые работают подобным образом:

<code>\sum</code>	$\Sigma$	<code>\prod</code>	$\Pi$	<code>\coprod</code>	$\coprod$
<code>\bigoplus</code>	$\bigoplus$	<code>\bigotimes</code>	$\bigotimes$	<code>\bigodot</code>	$\bigodot$
<code>\bigcup</code>	$\bigcup$	<code>\bigcap</code>	$\bigcap$	<code>\biguplus</code>	$\biguplus$
<code>\bigsqcup</code>	$\bigsqcup$	<code>\bigvee</code>	$\bigvee$	<code>\bigwedge</code>	$\bigwedge$

<code>\int</code>	$\int$	<code>\oint</code>	$\oint$	<code>\iint<sup>[1]</sup></code>	$\iint$
<code>\iiint<sup>[1]</sup></code>	$\iiint$	<code>\iiint<sup>[1]</sup></code>	$\iiint$	<code>\idotsint<sup>[1]</sup></code>	$\int\cdots\int$

Что бы использоваться больше интегральных символов, которые не входят в шрифт Computer Modern, подключите пакет `\[ \int\limits_a^b\frac{1}{(1+x)^{-3/2}}\,\mathrm{d}x= \left.-\frac{1}{\sqrt{1+x}}\right|_a^b \]`.

Команда `\substack[1]` позволяет использовать два обратных слэша `\\` для записи пределов Ряда или интеграла в несколько строк:

<pre>\sum_{\substack{0&lt;i&lt;m\\0&lt;j&lt;n}}P(i,j)</pre>	$\sum_{\substack{0<i<m\\0<j<n}}P(i,j)$
---	--

Если вы хотите, чтобы пределы интеграла были указаны выше или ниже символа, используйте команду `\limits`:

<pre>\int\limits_a^b</pre>	$\int_a^b$
----------------------------	------------

Однако если вы хотите, чтобы это применялось ко ВСЕМ интегралам, то нужно указать параметр `Шаблон:LaTeX/Parameter` при подключении пакета `\[ \int\limits_a^b\frac{1}{(1+x)^{-3/2}}\,\mathrm{d}x= \left.-\frac{1}{\sqrt{1+x}}\right|_a^b \]`: `Шаблон:LaTeX/Usage`

Нижние и верхние пределы в других контекстах, а также другие параметры пакета `\[ \int\limits_a^b\frac{1}{(1+x)^{-3/2}}\,\mathrm{d}x= \left.-\frac{1}{\sqrt{1+x}}\right|_a^b \]` связанные с ними, описаны в главе [Advanced Mathematics](#).

Для больших интегралов можно использовать пакет `\[ \int\limits_a^b\frac{1}{(1+x)^{-3/2}}\,\mathrm{d}x= \left.-\frac{1}{\sqrt{1+x}}\right|_a^b \]` <sup>[2]</sup>.

## Скобки, фигурные скобки и разделители

Как использовать фигурные скобки в многострочных уравнениях описано в главе [Advanced Mathematics](#).

Использование разделителей, таких как скобки, становится важным при работе с чем угодно, кроме самых тривиальных уравнений. Без них формулы могут стать двусмысленными. Кроме того, специальные типы математических структур, такие как матрицы, невозможно представить без разделителей.

Существует множество разделителей, доступных для использования в LaTeX:

<pre>( a ), [ b ], \{ c \},   d  , \l e \l, \angle f \angle, \lfloor g \rfloor, \lceil h \rceil, \ulcorner i \urcorner, / j \backslash</pre>	$(a), [b], \{c\},  d , \ e\ , \langle f \rangle, [g], [h], \lceil i \rceil, /j \backslash$
--	--

где `\lbrack` и `\rbrack` могут использоваться вместо "[" и "]".

### Автоматическое определение размеров

Очень часто математические функции будут отличаться друг от друга по размеру, и в этом случае разделители, окружающие выражение, должны изменяться соответственно. Это можно сделать автоматически с помощью команд `\left`, `\right`, и `\middle`. Любой из выше перечисленных разделителей может быть использован в сочетании с этими командами:

<pre>\left(\frac{x^2}{y^3}\right)</pre>	$\left(\frac{x^2}{y^3}\right)$
---	--------------------------------

```
P\left(A=2\middle|\frac{A^2}{B}>4\right)
```

$$P\left(A=2\middle|\frac{A^2}{B}>4\right)$$

Фигурные скобки определяются иначе, с помощью `\left\{` и `\right\}`,

```
\left\{\frac{x^2}{y^3}\right\}
```

$$\left\{\frac{x^2}{y^3}\right\}$$

Если разделитель нужен только с одной стороны выражения, то с другой стороны разделить может быть обозначен точкой (`.`), что делает его невидимым.

```
\left.\frac{x^3}{3}\right|_0^1
```

$$\left.\frac{x^3}{3}\right|_0^1$$

### Ручное определение размеров

В некоторых случаях автоматический размер, создаваемый командами `\left` и `\right`, может быть отличен от того, что вы ожидаете увидеть, или вам требуется более точно контролировать размеры разделителей. В этом случае могут использоваться команды-модификаторы `\big`, `\Big`, `\bigg` и `\Bigg`:

```
( \big( \Big( \bigg( \Bigg(
```

$$(((($$

Эти команды в первую очередь полезны при работе с вложенными разделителями. Например, при наборе текста:

```
\frac{\mathrm d}{\mathrm d x} \left( k g(x) \right)
```

$$\frac{\mathrm d}{\mathrm d x} (kg(x))$$

Мы видим, что команды `\left` и `\right` создают разделители такого же размера, что и вложенные в них. Это может создать трудности при чтении записей. Чтобы исправить это, мы пишем:

```
\frac{\mathrm d}{\mathrm d x} \big( k g(x) \big)
```

$$\frac{\mathrm d}{\mathrm d x} (kg(x))$$

Ручное определение размеров также может быть полезно, когда уравнение слишком велико, поэтому заканчивается в конце страницы, и должно быть разделено на две строки с помощью команды `"align"`. Хотя команды `\left.` и `\right.` можно использовать для автоматического определения размеров разделителей на каждой строке, это может привести к неправильным размерам. Кроме того, нужно использовать ручное определение размеров, чтобы избежать чрезмерно больших разделителей, если между ними появляется `\underbrace` или аналогичная команда.

## Matrices and arrays

A basic matrix may be created using the environment<sup>[1]</sup>: in common with other table-like structures, entries are specified by row, with columns separated using an ampersand (`&`) and a new rows separated with a double backslash (`\\`)

```
\[
\begin{matrix}
a & b & c \\
d & e & f \\
g & h & i
\end{matrix}
\]
```

$$\begin{matrix} a & b & c \\ d & e & f \\ g & h & i \end{matrix}$$



```
\begin{matrix}
-1 & 3 \\
2 & -4
\end{matrix}
=
\begin{matrix*}[r]
-1 & 3 \\
2 & -4
\end{matrix*}
```

$$\begin{matrix} -1 & 3 \\ 2 & -4 \end{matrix} = \begin{matrix} -1 & 3 \\ 2 & -4 \end{matrix}$$

The alignment by default is `\left` but it can be any column type valid in environment.

However matrices are usually enclosed in delimiters of some kind, and while it is possible to use the `\left` and `\right` commands, there are various other predefined environments which automatically include delimiters:

Environment name	Surrounding delimiter	Notes
[1]	()	centers columns by default
[3]	()	allows to specify alignment of columns in optional parameter
[1]	[]	centers columns by default
[3]	[]	allows to specify alignment of columns in optional parameter
[1]	{ }	centers columns by default
[3]	{ }	allows to specify alignment of columns in optional parameter
[1]		centers columns by default
[3]		allows to specify alignment of columns in optional parameter
[1]		centers columns by default
[3]		allows to specify alignment of columns in optional parameter

When writing down arbitrary sized matrices, it is common to use horizontal, vertical and diagonal triplets of dots (known as ellipses) to fill in certain columns and rows. These can be specified using the `\cdots`, `\vdots` and `\ddots` respectively:

```
A_{m,n} =
\begin{pmatrix}
a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\
a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\
\vdots & \vdots & \ddots & \vdots \\
a_{m,1} & a_{m,2} & \cdots & a_{m,n}
\end{pmatrix}
```

$$A_{m,n} = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{pmatrix}$$

In some cases you may want to have finer control of the alignment within each column, or want to insert lines between columns or rows. This can be achieved using the environment, which is essentially a math-mode version of the `\table` environment, which requires that the columns be pre-specified:

```
\begin{array}{c|c}
1 & 2 \\ \hline
3 & 4
\end{array}
```

$$\begin{array}{c|c} 1 & 2 \\ \hline 3 & 4 \end{array}$$

You may see that the AMS matrix class of environments doesn't leave enough space when used together with fractions resulting in output similar to this:

$$M = \begin{bmatrix} \frac{5}{6} & \frac{1}{6} & 0 \\ \frac{5}{6} & 0 & \frac{1}{6} \\ 0 & \frac{5}{6} & \frac{1}{6} \end{bmatrix}$$

To counteract this problem, add additional leading space with the optional parameter to the `\left` command:

```
M = \begin{bmatrix}
\frac{5}{6} & \frac{1}{6} & 0 \\
\frac{5}{6} & 0 & \frac{1}{6} \\
0 & \frac{5}{6} & \frac{1}{6}
\end{bmatrix}
```

$$M = \begin{bmatrix} \frac{5}{6} & \frac{1}{6} & 0 \\ \frac{5}{6} & 0 & \frac{1}{6} \\ 0 & \frac{5}{6} & \frac{1}{6} \end{bmatrix}$$

If you need "border" or "indexes" on your matrix, plain TeX provides the macro `\bordermatrix`

```
M = \bordermatrix{~ & x & y \cr
A & 1 & 0 \cr
B & 0 & 1 \cr}
```

$$M = \begin{matrix} & \begin{matrix} x & y \end{matrix} \\ \begin{matrix} A \\ B \end{matrix} & \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \end{matrix}$$

## Matrices in running text

To insert a small matrix, and not increase leading in the line containing it, use environment:

```
A matrix in text must be set smaller:
 $\bigl(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\bigr)$ 
to not increase leading in a portion of
text.
```

A matrix in text must be set smaller:  $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$  to not increase leading in a portion of text.

## Adding text to equations

The math environment differs from the text environment in the representation of text. Here is an example of trying to represent text within the math environment:

```
50 apples \times 100 apples = lots of apples^2
```

$$50\textit{apples} \times 100\textit{apples} = \textit{lots of apples}^2$$

There are two noticeable problems: there are no spaces between words or numbers, and the letters are italicized and more spaced out than normal. Both issues are simply artifacts of the maths mode, in that it treats it as a mathematical expression: spaces are ignored (LaTeX spaces mathematics according to its own rules), and each character is a separate element (so are not positioned as closely as normal text).

There are a number of ways that text can be added properly. The typical way is to wrap the text with the `\text{...}` command <sup>[1]</sup> (a similar command is `\mbox{...}`, though this causes problems with subscripts, and has a less descriptive name). Let's see what happens when the above equation code is adapted:

```
50 \text{apples} \times 100 \text{apples}
= \text{lots of apples}^2
```

$$50\text{apples} \times 100\text{apples} = \text{lots of apples}^2$$

The text looks better. However, there are no gaps between the numbers and the words. Unfortunately, you are required to explicitly add these. There are many ways to add spaces between maths elements, but for the sake of simplicity we may simply insert space characters into the `\text` commands.

```
50 \text{ apples} \times 100 \text{ apples}
= \text{lots of apples}^2
```

$$50 \text{ apples} \times 100 \text{ apples} = \text{lots of apples}^2$$

## Formatted text

Using the `\text` is fine and gets the basic result. Yet, there is an alternative that offers a little more flexibility. You may recall the introduction of font formatting commands, such as `\textrm`, `\textit`, `\textbf`, etc. These commands format the argument accordingly, e.g., `\textbf{bold text}` gives **bold text**. These commands are equally valid within a maths environment to include text. The added benefit here is that you can have better control over the font formatting, rather than the standard text achieved with `\text`.

```
50 \textrm{ apples} \times 100
\textbf{ apples} = \textit{lots of apples}^2
```

$50 \text{ apples} \times 100 \text{ apples} = \textit{lots of apples}^2$

Formatting mathematics symbols

See also: *w:Mathematical Alphanumeric Symbols*, *w:Help:Displaying a formula#Alphabets and typefaces* and *w:Wikipedia:LaTeX symbols#Fonts*

We can now format text; what about formatting mathematical expressions? There are a set of formatting commands very similar to the font formatting ones just used, except that they are specifically aimed at text in math mode (requires `\[ \int\limits_a^b\frac{1}{2} (1+x)^{-3/2}\, \mathrm{d}x= \left.-\frac{1}{\sqrt{1+x}} \right|_a^b \]`)

LaTeX command	Sample	Description	Common use
<code>\mathnormal{...}</code> (or simply omit any command)	<i>ABCDEF abcdef 123456</i>	The default math font	Most mathematical notation
<code>\mathrm{...}</code>	<b>ABCDEF abcdef 123456</b>	This is the default or normal font, unitalicised	Units of measurement, one word functions
<code>\mathit{...}</code>	<i>ABCDEF abcdef 123456</i>	Italicised font	Multi-letter function or variable names. Compared to <code>\mathnormal</code> , words are spaced more naturally and numbers are italicized as well.
<code>\mathbf{...}</code>	<b>ABCDEF abcdef 123456</b>	Bold font	Vectors
<code>\mathsf{...}</code>	<b>ABCDEF abcdef 123456</b>	<u>Sans-serif</u>	Categories
<code>\mathtt{...}</code>	<b>ABCDEF abcdef 123456</b>	<u>Monospace (fixed-width) font</u>	
<code>\mathfrak{...}</code>	<b>ABCDEF abcdef 123456</b>	<u>Fraktur</u>	Almost canonical font for Lie algebras, <u>ideals</u> in ring theory
<code>\mathcal{...}</code>	<i>ABCDEF</i>	Calligraphy (uppercase only)	Often used for sheaves/schemes and categories, used to denote cryptological concepts like an <i>alphabet of definition</i> ( <i>A</i> ), <i>message space</i> ( <i>M</i> ), <i>ciphertext space</i> ( <i>C</i> ) and <i>key space</i> ( <i>K</i> ); Kleene's <i>O</i> ; naming convention in description logic; Laplace transform ( <i>L</i> ) and Fourier transform ( <i>F</i> )
<code>\mathbb{...}</code> (requires the <code>\int\limits_a^b\frac{1}{2} (1+x)^{-3/2}\, \mathrm{d}x= \left.-\frac{1}{\sqrt{1+x}} \right _a^b \]</code> or <code>\int\limits_a^b\frac{1}{2} (1+x)^{-3/2}\, \mathrm{d}x= \left.-\frac{1}{\sqrt{1+x}} \right _a^b \]</code> package)	<b>ABCDEF</b>	<u>Blackboard bold (uppercase only)</u>	Used to denote special sets (e.g. real numbers)
<code>\mathscr{...}</code> (requires the <code>\int\limits_a^b\frac{1}{2} (1+x)^{-3/2}\, \mathrm{d}x= \left.-\frac{1}{\sqrt{1+x}} \right _a^b \]</code> package)	<i>ABCDEF</i>	<u>Script (uppercase only)</u>	An alternative font for categories and sheaves.

These formatting commands can be wrapped around the entire equation, and not just on the textual elements: they only format letters, numbers, and uppercase Greek, and other math commands are unaffected.

To bold lowercase Greek or other symbols use the `\boldsymbol` command<sup>[1]</sup>; this will only work if there exists a bold version of the symbol in the current font. As a last resort there is the `\pmb` command<sup>[1]</sup> (poor man's bold): this prints multiple versions of the character slightly offset against each other.

`\boldsymbol{\beta} = (\beta_1, \beta_2, \dotsc, \beta_n)`

$\beta = (\beta_1, \beta_2, \dots, \beta_n)$

To change the size of the fonts in math mode, see [\[../Advanced Mathematics#Changing font size|Changing font size\]](#).

Accents

So what to do when you run out of symbols and fonts? Well, the next step is to use accents:

<code>a' or a^{\prime}</code>	$a'$	<code>a''</code>	$a''$
<code>\hat{a}</code>	$\hat{a}$	<code>\bar{a}</code>	$\bar{a}$
<code>\grave{a}</code>	$\grave{a}$	<code>\acute{a}</code>	$\acute{a}$
<code>\dot{a}</code>	$\dot{a}$	<code>\ddot{a}</code>	$\ddot{a}$
<code>\not{a}</code>	$\not{a}$	<code>\mathring{a}</code>	$\mathring{a}$
<code>\overrightarrow{AB}</code>	$\overrightarrow{AB}$	<code>\overleftarrow{AB}</code>	$\overleftarrow{AB}$
<code>a'''</code>	$a'''$	<code>a''''</code>	$a''''$
<code>\overline{aaa}</code>	$\overline{aaa}$	<code>\check{a}</code>	$\check{a}$
<code>\breve{a}</code>	$\breve{a}$	<code>\vec{a}</code>	$\vec{a}$
<code>\dddot{a}^{[1]}</code>	$\dddot{a}^{[1]}$	<code>\ddddot{a}^{[1]}</code>	$\ddddot{a}^{[1]}$
<code>\widehat{AAA}</code>	$\widehat{AAA}$	<code>\widetilde{AAA}</code>	$\widetilde{AAA}$
<code>\stackrel{\frown}{AAA}</code>	$\stackrel{\frown}{AAA}$		
<code>\tilde{a}</code>	$\tilde{a}$	<code>\underline{a}</code>	$\underline{a}$

Color

The package `\[ \int\limits_a^b\frac{1}{(1+x)^{-3/2}}\,dx = \left.-\frac{1}{\sqrt{1+x}}\right|_a^b \]`, described in [\[../Colors#Adding\\_the\\_color\\_package|Colors\]](#), allows us to add color to our equations. For example,

`k = {\color{red}x} \mathbin{\color{blue}-} 2`

$k = x-2$

The only problem is that this disrupts the default `Шаблон:LaTeX` formatting around the `-` operator. To fix this, we enclose it in a `\mathbin` environment, since `-` is a binary operator. This process is described [here](http://tex.stackexchange.com/questions/21598/how-to-color-math-symbols) (<http://tex.stackexchange.com/questions/21598/how-to-color-math-symbols>).

Plus and minus signs

LaTeX deals with the `+` and `-` signs in two possible ways. The most common is as a binary operator. When two maths elements appear on either side of the sign, it is assumed to be a binary operator, and as such, allocates some space to either side of the sign. The alternative way is a sign designation. This is when you state whether a mathematical quantity is either positive or negative. This is common for the latter, as in math, such elements are assumed to be positive unless a `-` is prefixed to it. In this instance, you want the sign to appear close to the appropriate element to show their association. If you put a `+` or a `-` with nothing before it but you want it to be handled like a binary operator you can add an *invisible* character before the operator using `{}`. This can be useful if you are writing multiple-line formulas, and a new line could start with a `-` or `+`, for example, then you can fix some strange alignments adding the invisible character where necessary.

A plus-minus sign is written as:

`\pm`

$\pm$

Similarly, there exists also a minus-plus sign:

`\mp`

$\mp$

Controlling horizontal spacing

LaTeX is obviously pretty good at typesetting maths—it was one of the chief aims of the core TeX system that LaTeX extends. However, it can't always be relied upon to accurately interpret formulas in the way you did. It has to make certain assumptions when there are ambiguous expressions. The result tends to be slightly incorrect horizontal spacing. In these events, the output is still satisfactory, yet any perfectionists will no doubt wish to *fine-tune* their formulas to ensure spacing is correct. These are generally very subtle adjustments.

There are other occasions where LaTeX has done its job correctly, but you just want to add some space, maybe to add a comment of some kind. For example, in the following equation, it is preferable to ensure there is a decent amount of space between the maths and the text.

```
\[ f(n) =
\begin{cases}
n/2 & \text{if } n \text{ is even} \\
-(n+1)/2 & \text{if } n \text{ is odd}
\end{cases}
\]
```

$$f(n) = \begin{cases} n/2 & \text{if } n \text{ is even} \\ -(n+1)/2 & \text{if } n \text{ is odd} \end{cases}$$

This code produces errors with Miktex 2.9 and does not yield the results seen on the right. Use `\mathrm` instead of just `\text`.

(Note that this particular example can be expressed in more elegant code by the construct provided by the `\int\limits_a^b\frac{1}{(1+x)^{-3/2}}`, `\mathrm{d}x= \left.-\frac{1}{\sqrt{1+x}} \right|_a^b` package described in [Advanced Mathematics](#) chapter.)

LaTeX has defined two commands that can be used anywhere in documents (not just maths) to insert some horizontal space. They are `\quad` and `\qquad`

A `\quad` is a space equal to the current font size. So, if you are using an 11pt font, then the space provided by `\quad` will also be 11pt (horizontally, of course.) The `\qquad` gives twice that amount. As you can see from the code from the above example, `\quad`s were used to add some separation between the maths and the text.

OK, so back to the fine tuning as mentioned at the beginning of the document. A good example would be displaying the simple equation for the indefinite integral of  $y$  with respect to  $x$ :

$$\int y \, dx$$

If you were to try this, you may write:

```
\int y \mathrm{d}x
```

$$\int y dx$$

However, this doesn't give the correct result. LaTeX doesn't respect the white-space left in the code to signify that the  $y$  and the  $dx$  are independent entities. Instead, it lumps them altogether. A `\quad` would clearly be overkill in this situation—what is needed are some small spaces to be utilized in this type of instance, and that's what LaTeX provides:

Command	Description	Size
<code>\,</code>	small space	3/18 of a quad
<code>\:</code>	medium space	4/18 of a quad
<code>\;</code>	large space	5/18 of a quad
<code>\!</code>	negative space	-3/18 of a quad

NB you can use more than one command in a sequence to achieve a greater space if necessary.

So, to rectify the current problem:

```
\int y\, \mathrm{d}x
```

$$\int y \, dx$$

```
\int y\: \mathrm{d}x
```

$$\int y \, dx$$

```
\int y\; \mathrm{d}x
```

$$\int y \, dx$$

The negative space may seem like an odd thing to use, however, it wouldn't be there if it didn't have *some* use! Take the following example:

```
\left(
  \begin{array}{c}
    n \\
    r
  \end{array}
\right) = \frac{n!}{r!(n-r)!}
```

$$\binom{n}{r} = \frac{n!}{r!(n-r)!}$$

The matrix-like expression for representing binomial coefficients is too padded. There is too much space between the brackets and the actual contents within. This can easily be corrected by adding a few negative spaces after the left bracket and before the right bracket.

```
\left(\!
  \begin{array}{c}
    n \\
    r
  \end{array}
\!\right) = \frac{n!}{r!(n-r)!}
```

$$\binom{n}{r} = \frac{n!}{r!(n-r)!}$$

In any case, adding some spaces manually should be avoided whenever possible: it makes the source code more complex and it's against the basic principles of a What You See is What You Mean approach. The best thing to do is to define some commands using all the spaces you want and then, when you use your command, you don't have to add any other space. Later, if you change your mind about the length of the horizontal space, you can easily change it modifying only the command you defined before. Let us use an example: you want the *d* of a *dx* in an integral to be in roman font and a small space away from the rest. If you want to type an integral like `\int x \, \mathrm{d} x`, you can define a command like this: [Шаблон:LaTeX/Usage](#) in the preamble of your document. We have chosen `\dd` just because it reminds the "d" it replaces and it is fast to type. Doing so, the code for your integral becomes `\int x \, \dd x`. Now, whenever you write an integral, you just have to use the `\dd` instead of the "d", and all your integrals will have the same style. If you change your mind, you just have to change the definition in the preamble, and all your integrals will be changed accordingly.

## Manually Specifying Formula Style

To manually display a fragment of a formula using text style, surround the fragment with curly braces and prefix the fragment with `\textstyle`. The braces are required because the `\textstyle` macro changes the state of the renderer, rendering all subsequent mathematics in text style. The braces limit this change of state to just the fragment enclosed within. For example, to use text style for just the summation symbol in a sum, one would enter [Шаблон:LaTeX/Usage](#) The same thing as a command would look like this: [Шаблон:LaTeX/Usage](#) Note the extra braces. Just one set around the expression won't be enough. That would cause all math after `\tsum` *k* to be displayed using text style.

To display part of a formula using display style, do the same thing, but use `\displaystyle` instead.

## Advanced Mathematics: AMS Math package

The AMS (American Mathematical Society) mathematics package is a powerful package that creates a higher layer of abstraction over mathematical LaTeX language; if you use it it will make your life easier. Some commands `\int\limits_a^b\frac{1}{(1+x)^{-3/2}}\mathrm{d}x= \left.-\frac{1}{\sqrt{1+x}}\right|_a^b \]` introduces will make other plain LaTeX commands obsolete: in order to keep consistency in the final output you'd better use `\int\limits_a^b\frac{1}{(1+x)^{-3/2}}\mathrm{d}x= \left.-\frac{1}{\sqrt{1+x}}\right|_a^b \]` commands whenever possible. If you do so, you will get an elegant output without worrying about alignment and other details, keeping your source code readable. If you want to use it, you have to add this in the preamble: [Шаблон:LaTeX/Usage](#)

### Introducing dots in formulas

`\[ \int\limits_a^b\frac{1}{\sqrt{1+x}}\mathrm{d}x= \left.-\frac{1}{\sqrt{1+x}}\right|_a^b \]` defines also the `\dots` command, that is a generalization of the existing `\ldots`. You can use `\dots` in both text and math mode and LaTeX will replace it with three dots "..." but it will decide according to the context whether to put it on the bottom (like `\ldots`) or centered (like `\cdots`).

**Dots**

LaTeX gives you several commands to insert dots (ellipses) in your formulae. This can be particularly useful if you have to type big matrices omitting elements. First of all, here are the main dots-related commands LaTeX provides:

Code	Output	Comment
<code>\dots</code>	...	generic dots (ellipsis), to be used in text (outside formulae as well). It automatically manages whitespaces before and after itself according to the context, it's a higher level command.
<code>\ldots</code>	...	the output is similar to the previous one, but there is no automatic whitespace management; it works at a lower level.
<code>\cdots</code>	...	These dots are centered relative to the height of a letter. There is also the binary multiplication operator, <code>\cdot</code> , mentioned below.
<code>\vdots</code>	⋮	vertical dots
<code>\ddots</code>	⋮	diagonal dots
<code>\iddots</code>		inverse diagonal dots (requires the <code>\int\limits_a^b\frac{1}{\sqrt{1+x}}\mathrm{d}x= \left.-\frac{1}{\sqrt{1+x}}\right _a^b</code> package)
<code>\hdotsfor{n}</code>	.....	to be used in matrices, it creates a row of dots spanning <i>n</i> columns.

Instead of using `\ldots` and `\cdots`, you should use the semantically oriented commands. It makes it possible to adapt your document to different conventions on the fly, in case (for example) you have to submit it to a publisher who insists on following house tradition in this respect. The default treatment for the various kinds follows American Mathematical Society conventions.

Code	Output	Comment
<code>A_1,A_2,\dotsc,</code>	$A_1,A_2,\dots,$	for "dots with commas"
<code>A_1+\dotsb+A_N</code>	$A_1+\dots+A_N$	for "dots with binary operators/relations"
<code>A_1\dotsm A_N</code>	$A_1\dots A_N$	for "multiplication dots"
<code>\int_a^b \dotsi</code>	$\int_a^b \dots$	for "dots with integrals"
<code>A_1\dotso A_N</code>	$A_1\dots A_N$	for "other dots" (none of the above)

Write an equation with the align environment

How to write an equation with the align environment with the `\int\limits_a^b\frac{1}{\sqrt{1+x}}\mathrm{d}x= \left.-\frac{1}{\sqrt{1+x}}\right|_a^b` package is described in [Advanced Mathematics](#).

List of mathematical symbols

All the pre-defined mathematical symbols from the `\TeX\` package are listed below. More symbols are available from extra packages.

Relation Symbols									
Symbol	Script	Symbol	Script	Symbol	Script	Symbol	Script	Symbol	Script
<	<	>	>	=	=	∥	\parallel	⋈	\nparallel
≤	\leq	≥	\geq	≐	\doteq	×	\asymp	⋈	\bowtie
≪	\ll	≫	\gg	≐	\equiv	⊢	\vdash	⊢	\dashv
⊂	\subset	⊃	\supset	≈	\approx	∈	\in	∋	\ni
⊆	\subseteq	⊇	\supseteq	≅	\cong	⌢	\smile	⌣	\frown
⊈	\nsubseteq	⊉	\nsupseteq	≈	\simeq	⊨	\models	⊄	\notin
⊊	\sqsubset	⊋	\sqsupset	~	\sim	⊥	\perp		\mid
⊋	\sqsupseteq	⊌	\sqsubseteq	α	\propto	⌵	\prec	⌵	\succ
⌵	\preceq	⌶	\succeq	≠	\neq	∠	\sphericalangle	∠	\measuredangle
∴	\therefore	∵	\because						

Binary Operations

Symbol	Script	Symbol	Script	Symbol	Script	Symbol	Script
$\pm$	<code>\pm</code>	$\cap$	<code>\cap</code>	$\diamond$	<code>\diamond</code>	$\oplus$	<code>\oplus</code>
$\mp$	<code>\mp</code>	$\cup$	<code>\cup</code>	$\Delta$	<code>\bigtriangleup</code>	$\ominus$	<code>\ominus</code>
$\times$	<code>\times</code>	$\uplus$	<code>\uplus</code>	$\nabla$	<code>\bigtriangledown</code>	$\otimes$	<code>\otimes</code>
$\div$	<code>\div</code>	$\sqcap$	<code>\sqcap</code>	$\triangleleft$	<code>\triangleleft</code>	$\oslash$	<code>\oslash</code>
$*$	<code>\ast</code>	$\sqcup$	<code>\sqcup</code>	$\triangleright$	<code>\triangleright</code>	$\odot$	<code>\odot</code>
$\star$	<code>\star</code>	$\vee$	<code>\vee</code>	$\bigcirc$	<code>\bigcirc</code>	$\circ$	<code>\circ</code>
$\dagger$	<code>\dagger</code>	$\wedge$	<code>\wedge</code>	$\bullet$	<code>\bullet</code>	$\setminus$	<code>\setminus</code>
$\ddagger$	<code>\ddagger</code>	$\cdot$	<code>\cdot</code>	$\wr$	<code>\wr</code>	$\amalg$	<code>\amalg</code>

Set and/or Logic Notation

Symbol	Script	Symbol	Script
$\exists$	<code>\exists</code>	$\rightarrow$	<code>\rightarrow</code> or <code>\to</code>
$\nexists$	<code>\nexists</code>	$\leftarrow$	<code>\leftarrow</code> or <code>\gets</code>
$\forall$	<code>\forall</code>	$\mapsto$	<code>\mapsto</code>
$\neg$	<code>\neg</code>	$\implies$	<code>\implies</code>
$\subset$	<code>\subset</code>	$\impliedby$	<code>\impliedby</code>
$\supset$	<code>\supset</code>	$\Rightarrow$	<code>\Rightarrow</code> or <code>\implies</code>
$\in$	<code>\in</code>	$\leftrightarrow$	<code>\leftrightarrow</code>
$\notin$	<code>\notin</code>	$\iff$	<code>\iff</code>
$\ni$	<code>\ni</code>	$\Leftrightarrow$	<code>\Leftrightarrow</code> (preferred for equivalence (iff))
$\wedge$	<code>\wedge</code>	$\top$	<code>\top</code>
$\vee$	<code>\vee</code>	$\bot$	<code>\bot</code>
$\angle$	<code>\angle</code>	$\emptyset$ and $\varnothing$	<code>\emptyset</code> and <code>\varnothing</code> <sup>[1]</sup> ( <a href="https://ru.wikibooks.org/wiki/%D0%9C%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B5_%D1%84%D0%BE%D1%80%D0%BC%D1%83%D0%BB%D1%8B_%D0%B2_LaTeX#endnote_symbolpackage">https://ru.wikibooks.org/wiki/%D0%9C%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B5_%D1%84%D0%BE%D1%80%D0%BC%D1%83%D0%BB%D1%8B_%D0%B2_LaTeX#endnote_symbolpackage</a> )
		$\rightharpoonup$	<code>\rightharpoonup</code>

Delimiters

Symbol	Script	Symbol	Script	Symbol	Script	Symbol	Script
$ $	<code> </code> or <code>\mid</code> (difference in spacing)	$\ $	<code>\ </code>	$/$	<code>/</code>	$\backslash$	<code>\backslash</code>
$\{$	<code>\{</code>	$\}$	<code>\}</code>	$\langle$	<code>\langle</code>	$\rangle$	<code>\rangle</code>
$\uparrow$	<code>\uparrow</code>	$\Uparrow$	<code>\Uparrow</code>	$\lceil$	<code>\lceil</code>	$\rceil$	<code>\rceil</code>
$\downarrow$	<code>\downarrow</code>	$\Downarrow$	<code>\Downarrow</code>	$\lfloor$	<code>\lfloor</code>	$\rfloor$	<code>\rfloor</code>

Note: To use the Greek Letters in LaTeX that have the same appearance in the Latin alphabet, just use Latin: e.g., A instead of Alpha, B instead of Beta, etc.

Greek Letters

Symbol	Script	Symbol	Script
<b>A</b> and $\alpha$	A and <code>\alpha</code>	<b>N</b> and $\nu$	N and <code>\nu</code>
<b>B</b> and $\beta$	B and <code>\beta</code>	<b>Ξ</b> and $\xi$	<code>\Xi</code> and <code>\xi</code>
<b>Γ</b> and $\gamma$	<code>\Gamma</code> and <code>\gamma</code>	<b>O</b> and $o$	O and <code>o</code>
<b>Δ</b> and $\delta$	<code>\Delta</code> and <code>\delta</code>	<b>Π, π</b> and $\varpi$	<code>\Pi</code> , <code>\pi</code> and <code>\varpi</code>
<b>Ε, ε</b> and $\epsilon$	E, <code>\epsilon</code> and <code>\varepsilon</code>	<b>Ρ, ρ</b> and $\varrho$	P, <code>\rho</code> and <code>\varrho</code>
<b>Z</b> and $\zeta$	Z and <code>\zeta</code>	<b>Σ, σ</b> and $\varsigma$	<code>\Sigma</code> , <code>\sigma</code> and <code>\varsigma</code>
<b>Η</b> and $\eta$	H and <code>\eta</code>	<b>T</b> and $\tau$	T and <code>\tau</code>
<b>Θ, θ</b> and $\vartheta$	<code>\Theta</code> , <code>\theta</code> and <code>\vartheta</code>	<b>Υ</b> and $\upsilon$	<code>\Upsilon</code> and <code>\upsilon</code>
<b>I</b> and $\iota$	I and <code>\iota</code>	<b>Φ, φ</b> , and $\varphi$	<code>\Phi</code> , <code>\phi</code> and <code>\varphi</code>
<b>Κ, κ</b> and $\varkappa$	K, <code>\kappa</code> and <code>\varkappa</code>	<b>Χ</b> and $\chi$	X and <code>\chi</code>
<b>Λ</b> and $\lambda$	<code>\Lambda</code> and <code>\lambda</code>	<b>Ψ</b> and $\psi$	<code>\Psi</code> and <code>\psi</code>
<b>Μ</b> and $\mu$	M and <code>\mu</code>	<b>Ω</b> and $\omega$	<code>\Omega</code> and <code>\omega</code>



Other symbols

Symbol	Script		Symbol	Script		Symbol	Script		Symbol	Script
$\partial$	<code>\partial</code>		$\Re$	<code>\Re</code>		$\nabla$	<code>\nabla</code>		$\aleph$	<code>\aleph</code>
$\eth$	<code>\eth</code>		$\Im$	<code>\Im</code>		$\Box$	<code>\Box</code>		$\beth$	<code>\beth</code>
$\hbar$	<code>\hbar</code>		$\wp$	<code>\wp</code>		$\infty$	<code>\infty</code>		$\gimel$	<code>\gimel</code>

<sup>^</sup> Not predefined in LATEX 2. Use one of the packages `latexsym`, `amsfonts`, `amssymb`, `txfonts`, `pxfonts`, or `wasysym`

Trigonometric Functions

Symbol	Script	Symbol	Script	Symbol	Script	Symbol	Script
$\sin$	<code>\sin</code>	$\arcsin$	<code>\arcsin</code>	$\sinh$	<code>\sinh</code>	$\sec$	<code>\sec</code>
$\cos$	<code>\cos</code>	$\arccos$	<code>\arccos</code>	$\cosh$	<code>\cosh</code>	$\csc$	<code>\csc</code>
$\tan$	<code>\tan</code>	$\arctan$	<code>\arctan</code>	$\tanh$	<code>\tanh</code>		
$\cot$	<code>\cot</code>	$\operatorname{arccot}$	<code>\operatorname{arccot}</code>	$\coth$	<code>\coth</code>		

If LaTeX does not include a command for the mathematical operator you want to use, for example `\cis` (cosine plus i times sine), add to your preamble:

```
\DeclareMathOperator{\cis}{cis}
```

You can then use `\cis` in the document just like `\cos` or any other mathematical operator.

## Summary

As you begin to see, typesetting math can be tricky at times. However, because LaTeX provides so much control, you can get professional quality mathematics typesetting with relatively little effort (once you've had a bit of practice, of course!). It would be possible to keep going and going with math topics because it seems potentially limitless. However, with this tutorial, you should be able to get along sufficiently.

Шаблон:TODO

## Notes

- 1. requires the `\int\limits_a^b\frac{1}{2}(1+x)^{-3/2}\mathrm{d}x=\left.\frac{1}{\sqrt{1+x}}\right|_a^b` package
- 2. <http://hdl.handle.net/2268/6219>
- 3. requires the `\int\limits_a^b\frac{1}{2}(1+x)^{-3/2}\mathrm{d}x=\left.\frac{1}{\sqrt{1+x}}\right|_a^b` package

## Further reading

- [meta:Help:Displaying a formula](#): Wikimedia uses a subset of LaTeX commands.

## External links

- [detexify](http://detexify.kirelabs.org) (<http://detexify.kirelabs.org>): applet for looking up LaTeX symbols by drawing them
- [amsmath documentation](http://mirrors.ctan.org/macros/latex/required/amsmath/amsmath.pdf) (<http://mirrors.ctan.org/macros/latex/required/amsmath/amsmath.pdf>)
- [LaTeX - The Student Room](http://www.thestudentroom.co.uk/wiki/LaTeX) (<http://www.thestudentroom.co.uk/wiki/LaTeX>)
- [The Comprehensive LaTeX Symbol List](http://www.ctan.org/tex-archive/info/symbols/comprehensive) (<http://www.ctan.org/tex-archive/info/symbols/comprehensive>)
- [MathLex - LaTeX math translator and equation builder](http://mathlex.org/latex) (<http://mathlex.org/latex>)

### МЕТОДЫ ОЦЕНКИ КОМПЕТЕНТНОСТИ СТУДЕНТОВ

На сегодняшний день по-прежнему актуальной является проблема создания методов и технологий оценки компетентности студентов. Повсе-местное введение компетентностного подхода в российских вузах (в рам-ках действующих ГОС) не решило данную проблему. Разработка эффек-тивных методов и измерительных процедур оценки компетентности воз-можна только на основе методов математического моделирования и си-стемного анализа.

Компетентностный подход; высшее образование; модель компетент-ности студентов

Источник — [https://ru.wikibooks.org/w/index.php?title=Математические\\_формулы\\_в\\_LaTeX&oldid=208015](https://ru.wikibooks.org/w/index.php?title=Математические_формулы_в_LaTeX&oldid=208015)

Эта страница в последний раз была отредактирована 3 сентября 2021 в 12:06.

