

НЕМНОГО СОВЕТОВ ПО РАБОТЕ С РЕДАКТОРОМ МАТЕМАТИЧЕСКИХ ФОРМУЛ MATH ИЗ LIBREOFFICE

Ни для кого не секрет, что запись математических выражений ощутимо отличается от обычного текста. Чтобы оформить их подобающим образом при создании файла электронного документа (статьи, курсовой работы и т. п.) существует специальное программное обеспечение, к разряду которого относится приложение Math, входящее в состав пакета LibreOffice*.

Особенностью данной программы является то, что процесс редактирования формулы сопряжён с вводом её особого текстового описания, составляемого в соответствии с определёнными требованиями. Это описание («код») анализируется приложением и отображается им в «правильном», то есть в привычном нам, виде. Допустим, формула для нахождения корней квадратного уравнения

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

в нотации Math выглядит следующим образом:

$$x_{1,2} = \{ -b \pm \text{sqrt}\{ b^{\{2\}} - 4ac \} \} \text{ over } \{ 2a \}$$

Указанное обстоятельство повышает «порог вхождения» для работы с редактором, ведь для новичков такой подход к созданию математических формул крайне непривычен, но уверяю: при желании он довольно быстро осваивается и не вызывает особых затруднений. Даже если читающие сейчас эти строки не имели ранее дела с Math, то им вполне по силам разглядеть в приведённом выше примере кода формулы ряд его особенностей и сделать некоторые выводы об упомянутых правилах его составления. Первое, что бросается в глаза – символы фигурных скобок `{ }` играют важную служебную роль, логически отделяя какой-либо элемент математического выражения от остального его содержимого (например, нижний индекс обозначается знаком `"_"` (подчёркивание), а сразу за ним идёт то, что в этом самом индексе и надо написать – символы `"1,2"`). Также легко заключить, что есть целый ряд специальных слов, служащих цели «конструирования» формулы: `"over"` для дроби

`{ "числитель" } over { "знаменатель" }`

и `"sqrt"` для квадратного корня

`sqrt{ "подкоренное выражение" }`

В Сети есть большое число учебных пособий по Math, в настоящей же заметке мне хотелось бы рассказать лишь про некоторые особенности работы с этим приложением, основанные на личном опыте, поскольку именно данным редактором (как и другими программами из LibreOffice) я активно пользуюсь при подготовке своих интернет-публикаций**. Излагаемый далее материал в основном относится к windows-сборке пакета LibreOffice 7.6.6. Когда речь зайдёт об иной версии программы, то про это будет сказано отдельно.

1. Служебные слова

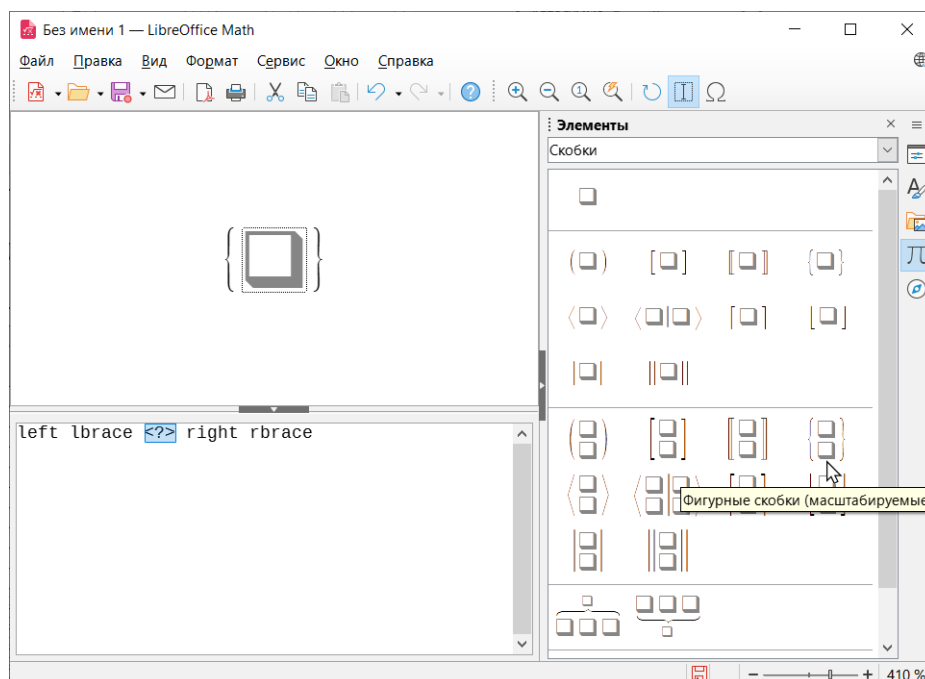
При работе в Math помнить служебные слова полезно, но не обязательно – типовые элементы формул собраны на специальной палитре шаблонов, при щелчке по ним в области ввода автоматически вставляется текст-заготовка кода, в котором далее нужно будет поменять заполнители

<?>

на нужную информацию:

* См. заметку «Microsoft Office Word и LibreOffice Writer (часть I)»
(URL: <https://shurichimik.narod.ru/consideration/19wordwriterp1/19word-and-writer-p1.htm#link20>).

** См. заметку «Мои сайт и дзен-канал: как это делается»
(URL: <https://shurichimik.narod.ru/consideration/40-toolkit-soft/40-toolkit-soft.htm>).



Однако есть служебное слово, которое, строго говоря, действительно надо знать, так как информация о его существовании отнюдь не на поверхности лежит. Слово это “none”, оно отменяет отображение одной из парных скобок – такое может понадобиться при записи системы уравнений:

Код	Отображение
<code>left lbrace binom{x+y=3}{x-y=1} right none</code>	$\begin{cases} x+y=3 \\ x-y=1 \end{cases}$

2. Структурирование кода при наборе

Правила составления кода формулы таковы, что когда она оказывается сложной, то в нём становится очень легко запутаться. Для повышения удобочитаемости стоит помнить: приложение в поле ввода игнорирует пробелы, переводы строки и символы табуляции. Это позволяет оформить код так, чтобы в нём было проще разобраться. Сравните:

Структурированный код	Неструктурированный код	Отображение
<pre>left [binom{ left lbrace binom{x=0}{y=1} right none }{ left lbrace binom{x=5}{y=7} right none } right none</pre>	<pre>left [binom{left lbrace binom{x=0}{y=1} right none} {left lbrace binom{x=5}{y=7} right none} right none</pre>	$\begin{cases} x=0 \\ y=1 \\ x=5 \\ y=7 \end{cases}$

Оба варианта кода отображаются в готовой формуле одинаково, но тот, который слева и где не свалено всё в кучу, понять легче.

3. Применение кавычек

Настоящей «палочкой-выручалочкой» в коде формулы служат кавычки

“ ”

Заключённые в них символы отображаются как есть, что и обуславливает целый ряд способов их использования.

3.1. Служебные символы

Как упоминалось выше, у фигурных скобок (а также у некоторых других знаков – у октоторпа #, например) имеется особое функциональное назначение, поэтому просто так их в формуле не напишешь, а потребность в этом может возникнуть – для обозначения дробной части числа, например. Для подобных случаев предусмотрены специальные служебные слова, но можно поступить иначе. Посмотрите на следующий код:

Код	Отображение
<code>lbrace x rbrace</code>	$\{x\}$
<code>"{"x"}"</code>	$\{x\}$

Лично мне больше по душе второй вариант, в первом фигурные скобки какие-то «худосочные» получаются.

3.2. Прямой шрифт

По правилам типографской вёрстки буквы, обозначающие переменные величины, набираются курсивом. Math следует этому правилу автоматически. Тем не менее иногда возникает необходимость отобразить буквы именно прямым шрифтом. Вот два примера, где кавычки пригодятся для этого.

а) словесное пояснение в формуле

Код	Отображение
<code>sqrt{"{"x"}"}=sqrt{"{"x+k"}"} ", где " k in setZ</code>	$\sqrt{\{x\}}=\sqrt{\{x+k\}}$, где $k \in \mathbb{Z}$ (периодичность функции квадратного корня из дробной части числа)

б) математическая выражение, в котором фигурируют химические формулы

Код	Отображение
<code>K_a = { "[C"_2 "O"_4^{2-}"] " cdot "[H"^{+}"]"^2 } over { alignc{"[H"_2 "C"_2 "O"_4]" } }</code>	$K_a = \frac{[C_2O_4^{2-}] \cdot [H^+]^2}{[H_2C_2O_4]}$ (константа диссоциации щавелевой кислоты)

3.3. Имена функций

Math «знает» написание многих типовых математических функций и при вводе их имён автоматически те распознаёт и отображает прямым шрифтом. Однако в отечественной и зарубежной литературе некоторые обозначения различаются. Тангенс числа x у нас принято писать как

$\operatorname{tg} x$

в отличие от международного варианта

$\tan x$

Мне привычнее наши «родные» обозначения, а поскольку LibreOffice в целом – софт иностранный, то для «русификации функций» я пользуюсь кавычками:

Код	Отображение
<code>tg x = 1 over { ctg x }</code>	$tg x = \frac{1}{ctg x}$
<code>"tg" x = 1 over { "ctg" x }</code>	$tg x = \frac{1}{ctg x}$

Такой подход пригождается, когда необходимо написать в формуле что-то, что вообще является лишь собственной выдумкой, такой как оператор «арифия» (среднее арифметическое значение функции на отрезке)*:

Код	Отображение
<code>"ariph" csub{[a; b]} f(x) ={ 1 } over { b-a } cdot int from{a} to{b} f(x) dx</code>	$arith_{[a;b]} f(x) = \frac{1}{b-a} \cdot \int_a^b f(x) dx$

3.4. Символы Юникода

Свои нюансы есть и при работе с символами, отсутствующими на стандартной компьютерной клавиатуре. В частности, для букв греческого алфавита можно использовать мнемоники общего вида

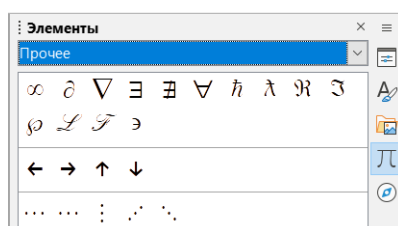
`%SymbolName`,

где `SymbolName` – название греческой буквы, набранное латиницей.

При потребности задействовать греческие буквы я предпочитаю иной способ: вставляю их непосредственно в код, так как Math умеет работать с символами Юникода. Не стоит забывать, что при использовании мнемоник приложение по умолчанию берёт изображения из гарнитуры “OpenSymbol”, поэтому начертание букв в этом случае может отличаться от варианта с прямой вставкой юникодовских знаков:

Код	Отображение
<code>sin(%alpha + %beta + %gamma)</code>	$\sin(\alpha + \beta + \gamma)$
<code>sin(α + β + γ)</code>	$\sin(\alpha + \beta + \gamma)$
<code>sin("α" + "β" + "γ")</code>	$\sin(\alpha + \beta + \gamma)$

Используя зарезервированные служебные слова можно вставлять в формулы и многие другие математические символы, такие как:



* См. заметку «“Теория средних” или Моё математическое графоманство» (URL: <https://shurichimik.narod.ru/consideration/17mediumtheory/17mediumtheory.htm/>)

Эти же знаки есть и в Юникоде, поэтому их также допустимо вставлять напрямую (обратите внимание на проявляющиеся при этом небольшие различия в начертании символов):

Код	Отображение
<pre>sqrt{ x^2 } = abs{ x } newline (forall x in setR)</pre>	$\sqrt{x^2} = x $ $(\forall x \in \mathbb{R})$
<pre>sqrt{ x^2 } = " x " newline ("∀" x "∈ℝ")</pre>	$\sqrt{x^2} = x $ $(\forall x \in \mathbb{R})$

Я рекомендую спецзнаки облачать в кавычки для правильного их отображения:

Код	Отображение
<pre>m_"⊕" over m_"☾" = 81,3</pre>	$\frac{m_{\oplus}}{m_{\ominus}} = 81,3$ <p>(отношение масс Земли \oplus и Луны \ominus)</p>

3.5. Отбивка пробелами для форматирования

При записи системы уравнений часто одно выражение оказывается короче другого, а содержимое масштабируемых фигурных скобок по умолчанию выравнивается по центру (а). Хотя в Math предусмотрена возможность для выравнивания объектов в формулах (б), при помощи пробелов, заключённых в кавычки, можно добиться аналогичного результата (в). По моему субъективному мнению такой способ несколько снижает визуальную «зашумлённость» кода служебными словами.

	Код	Отображение
(а)	<pre>left lbrace binom{ x <> y }{ 2 over { x-y } = x+y } right none</pre>	$\left\{ \begin{array}{l} x \neq y \\ \frac{2}{x-y} = x+y \end{array} \right.$
(б)	<pre>left lbrace binom{ alignl {x <> y} }{ 2 over { x-y } = x+y } right none</pre>	$\left\{ \begin{array}{l} x \neq y \\ \frac{2}{x-y} = x+y \end{array} \right.$

	Код	Отображение
(в)	<pre>left lbrace binom{ x <> y " " }{ 2 over { x-y } = x+y } right none</pre>	$\begin{cases} x \neq y \\ \frac{2}{x-y} = x+y \end{cases}$

4. Структурные формулы химических веществ

Выше (см. п. 3.2) уже упоминалось применение Math для записи выражений, относящихся к химии, но этим его возможности не ограничиваются. С помощью него можно изображать и некоторые другие несложные химические формулы. Вот несколько примеров.

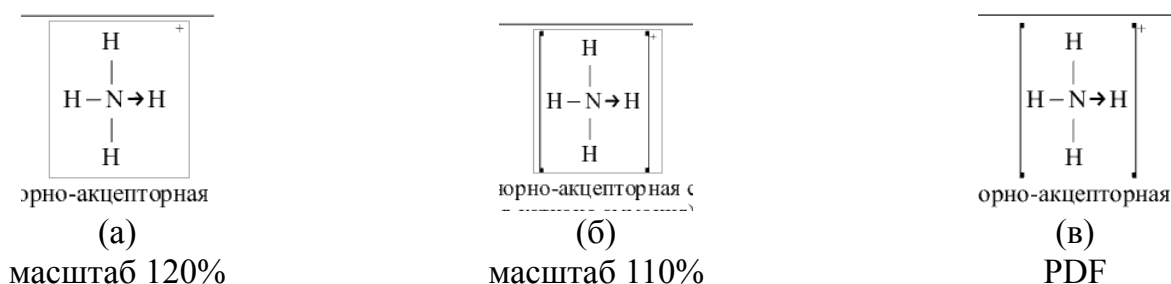
Код	Отображение
<pre>Stack{ " CH"_3 # " " slash "" # "O = C" # " " setminus "" # " CH"_3 }</pre>	$\begin{array}{c} \text{CH}_3 \\ / \\ \text{O} = \text{C} \\ \backslash \\ \text{CH}_3 \end{array}$ <p>(структурная формула ацетона)</p>
<pre>left [stack{ " H" # " " # "H"- "N" rightrightarrow "H" # " " # " H" } right]^{"+"}</pre>	$\left[\begin{array}{c} \text{H} \\ \\ \text{H} - \text{N} \rightarrow \text{H} \\ \\ \text{H} \end{array} \right]^+$ <p>(донорно-акцепторная связь в катионе аммония)</p>
<pre>stack{ alignc {"O" dotsaxis "H" - "O"} # alignc {" "} # "CH"_3 - "C C" - "CH"_3 # alignc {" "} # alignc {"O" - "H" dotsaxis "O"} }</pre>	$\begin{array}{ccc} & \text{O} \cdots \text{H} - \text{O} & \\ & \quad & \\ \text{CH}_3 - \text{C} & & \text{C} - \text{CH}_3 \\ & \quad & \\ & \text{O} - \text{H} \cdots \text{O} & \end{array}$ <p>(водородные связи в димере уксусной кислоты)</p>

5. Странности в работе редактора

В функционировании Math выявлены некоторые «шалости», проявляющиеся при отображении формул, вставленных в файл приложения офисного пакета (в текстовый документ Writer или в рисунок Draw). Открывать и редактировать эти файлы мне доводится на двух разных компьютерах и с чем конкретно связаны такие огрехи, я сказать затрудняюсь – вероятно дело в операционной системе (Windows 10 / Ubuntu 20.04.6 LTS) и/или в версии самого пакета LibreOffice (6.4.7 / 7.6.6).

5.1. Проблемы с отображением масштабируемых скобок

В случае компьютера с ОС Ubuntu, на которой установлен LibreOffice 6.4.7, при определённых значениях масштаба страницы открытого текстового документа Writer перестают показываться масштабируемые скобки в формуле, в этот документ вставленной (а). При уменьшении масштаба скобки появляются (б), при печати документа и при экспорте его в PDF они в выходном файле также присутствуют (в).



5.2. Некорректное отображение символа корня

Если файл редактировать на компьютере с Ubuntu / LibreOffice 6.4.7, а затем перенести его и открыть на компьютере с Windows / LibreOffice 7.6.6, то в формулах «съезжает» вбок горизонтальная палочка у корней:

$$c^2 = a^2 + b^2$$

этого равенства на $\sqrt{3}/4$, тогда

$$c^2 \cdot \frac{\sqrt{3}}{4} = a^2 \cdot \frac{\sqrt{3}}{4} + b^2 \cdot \frac{\sqrt{3}}{4}$$

Для исправления нужно сделать двойной щелчок по вставленной в документ Writer формуле. Она перейдёт в режим правки (при этом запустится Math), после чего надо выйти из этого режима, щёлкнув мышью в документе (за пределами габаритного поля формулы) и корень в формуле станет отображаться нормально. Аналогичный дефект возникает, если файл, созданный на компьютере с Windows / LibreOffice 7.6.6 перенести и открыть под Ubuntu / LibreOffice 6.4.7.

Если текстовый документ, некорректно отображающий корни в формулах, экспортировать в PDF, то палочки у корней в выходном файле также получатся «съехавшими».

5.3. Отображение символов Юникода

В п. 3.4 рекомендовалось при непосредственной вставке Юникод-знаков в код формулы «оборачивать» их кавычками, так как это позволяет добиться корректности и одинаковости изображения формулы в разных ОС и в разных версиях офисного пакета. Демонстрация того, что может произойти, если этого не сделать, приведена ниже на примере формулы отношения массы Земли к массе Луны с таким кодом:

$$m_{\oplus} \text{ over } m_{\lrcorner} = 81,3$$

ПО	Режим правки документа	Экспорт в PDF
Windows / LibreOffice 7.6.6	$\frac{m_{\square}}{m_{\zeta}} = 81,3$	$\frac{m_{\square}}{m_{\zeta}} = 81,3$
Ubuntu / LibreOffice 6.4.7	$\frac{m_{\oplus}}{m_{\zeta}} = 81,3$	$\frac{m_{\oplus}}{m_{\zeta}} = 81,3$

© Широков Александр, 24.09.2024