

# formul

для программистов

## 1 Общее описание

Скрипт `formul` предназначен для удобного перевода формул методистов в готовые формулы для генератора. Поддерживаются две основные функции: `exprtosol` и `exprtoans`, позволяющие переводить исходную формулу методиста в готовую для вставки её в какой-либо буфер и для вставка её и вычисления ответа соответственно.

Пример работы:

Исходная формула:  $-m_e + c \cdot v / k - c \cdot k^{\wedge}(v/c)$

```
exprtosol: |'-'|me|'+'|frac_|"cv"|_frac_|'k'|_frac_|
"-ck"|up_|frac_|'v'|_frac_|'c'|_frac_|up|
```

```
exprtoans: consts.fiz.c*Vv/kv-consts.fiz.c*
matreal.pow(kv,Vv/consts.fiz.c)-consts.fiz.me
```

## 2 Как начать использовать

### 2.1 разобраться с python3

Каждый уважающий себя человек должен обладать хотя бы минимальными знаниями этого языка. А для подобных целей преобразования одних текстовых данных в другие он просто не заменим. Легче повеситься, чем делать что-то подобное на чистых плюсах без каких-нибудь специальных библиотек для этого.

### 2.2 установить sympy

Под linux это должно быть как-то так

```
pip3 install sympy
```

Для всего остального есть

<http://docs.sympy.org/dev/install.html>

## 2.3 подготовить параметры

Для работы скрипта нужно подготовить таблицу замен. Она находится в двух файлах `param.txt` для `exprtans` и `paramsol.txt` для `exptosol`, которые должны находиться рядом с тем файлом, который вы выполняете. Примеры файлов:

`param.txt`

```
m_e consts.fiz.me
c consts.fiz.c
epsilon_{gamma} epsilongammav
k kv
v Vv
```

символы `{, }, \` игнорируются

Как видно можно использовать не только собственные переменные (`kv, Vv`), но и встроенные константы.

`paramsol.txt`

```
m_e me
c "c"
epsilon_{gamma} epsilongamma
k "k"
v "V"
a_1 |"a"|down1|
```

Для подстановки в решение можно использовать заранее заданные буферы (`epsilongamma, me`), создаваемые на лету (`a_1`) (их стоит окружить двумя вертикальными чертами) и просто безымянные переменные (`c, k, V`) (их стоит окружить двойными кавычками).

## 2.4 Краткое введение в python модули

Подключить этот скрипт можно дописав в начало вашего python файла одну из следующих строк:

1) `import formul`

в таком случае вызывать функции из него следует как `formul.exptosol()`

2) `from formul import *`

в таком случае вызывать функции из него следует просто `exptosol()`

Сам файл `formul.py` должен при этом быть или рядом с исполняемым файлом, или в одном из каталогов, из которого ваш `python` умеет брать библиотеки.

## 2.5 Всё как-то непонятно

Рабочий пример использования всегда можно найти в файле `tmp.py`. Также тестовые списки замен прилагаются. Я надеюсь, что файл `tmp.py` запустится на любой машине использующей третий питон и имеющей правильно установленный `sympy`. (Я этого не проверял). Если известно, что это не так, то стоит мне об этом сообщить.

## 3 Технические подробности

А что собственно под капотом?

Как уже упоминалось много раз для разбора выражения активно используется `sympy`, а точнее функция `parse_expr()` из него. Это хоть и накладывает некоторые ограничения на входящие формулы (которые, надеюсь, почти незаметны из-за предобработки выражения), но во много раз более надежны, чем тот велосипед, который я могу изобрести на коленке. После этого идет рекурсивное преобразование выражения.

### 3.1 Особенности

#### 3.1.1 `exprtoans`

Так как `sympy` всё равно разбирает выражение, то после этого оно максимально упрощается функцией `simplify()`, которая ищет запись этого выражения с наименьшим количеством действий. Ровно то, что и нужно. Стоит помнить, что `exprtoans` не распознаёт формулы, в которых есть знак равенства. (Непонятно как их интерпретировать, если слева не один символ. (если один символ, то может быть скоро прикручу))

#### 3.1.2 `exprtosol`

Для разбора выражения `parse_expr()` используется с `evaluate=False`. (Что пытается оставить запись методиста в первоизданном виде. Однако из-за бага в `sympy` (имеющего место быть, если выражение содержит "свои" функции типа  $F(a)$ ) это возможно не всегда. В таких случаях используется `evaluate=True`)

### 3.1.3 splitter

Маленьким бонусом идет функция `spliter`, которая из длинной входной строки(или строк разделенных `'\n'` возвращает (если может) строки пригодного размера для вставки в генератор. С нужными отступами. Вторым параметром подаётся список разделяющих элементов. (! Разделение глупое. Может порвать кавычки !).

Использование: `spliter(str,list('|'))`

### 3.1.4 Общие детали

В настоящий момент (и пока уходить от этой практики не планируется) все умножаемые компоненты выражения приводятся к общему виду  $\frac{a}{b}\sqrt{\frac{c}{d}}$ , даже если изначально методист записал его как  $\sqrt{c} \cdot a \cdot \frac{1}{b} \cdot \frac{1}{\sqrt{d}}$ . Это связано с тем, что с одной стороны при вычислении ответа корень из некоторых частей может и не извлекаться (так как что такое корень из метра неизвестно, а вот что такое корень из квадратного метра уже вполне понятно). С другой стороны и в формулах для решения совсем не факт, что методист хочет видеть что-то такое  $\frac{\sqrt{2}\sqrt{a}}{\sqrt{3}\sqrt{b}}$ . Его вполне устроит  $\sqrt{\frac{2a}{3b}}$ . Если где-то это неприемлемо, то, к сожалению, `formul` для этих случаев пока не поможет.

## 3.2 Пока нет, но может быть

- Поддержка неявного умножения.
- Поддержка разных доп.функций (`Root`, `Ln`,...). В настоящий момент есть только поддержка `Sqrt`.
- Исправление проблемы с символом `'S'`. Пока выражения с ним не распознаются
- ...

## 4 Агитация за использование

Скрипт, помимо очевидного прироста в скорости в преобразовании формул, а скрипт хоть и не самый производительный, но за пару секунд сотню формул переделает, что нельзя сказать о программисте, пьющем пятую кружку кофе, и постоянно забывающем какие-нибудь закрывающие `_frac`, да делающем опечатки в название переменных, даёт уже и

сейчас (а дальше больше), куда большую надежность, чем бедный вышеупомянутый программист. Уже сейчас он довольно неплохо справляется с большой частью формул.

## 5 Для связи

По всяким вопросам/предложениям/пожеланиям/замечаниям/ошибкам(и тухлым помидорам) стоит писать мне на почту: `retortakolba@yandex.ru` или в skype: `ser_polevoy`