

SugarTeX

SugarTeX is a more readable LaTeX language extension and a transcompiler to LaTeX.

See [PDF version of this documentation](#) - it nicely renders all Unicode characters. See original Markdown version [here](#).

Contents

- [Command line interfaces](#)
- [Tweaking SugarTeX](#)
- [SugarTeX replacements and operators](#)
 - [Math delimiters](#)
 - [New escape character](#)
 - [Brackets](#)
 - [Simple pre-replacements](#)
 - [Superscripts and Subscripts](#)
 - [Regular expressions pre-replacements](#)
 - [Nullary operators](#)
 - [Prefix unary operators](#)
 - [Styles](#)
 - [Styles with special brackets](#)
 - [Greedy prefix unary operators](#)
 - [Standard prefix unary operators](#)
 - [Postfix unary operators](#)

- Center binary operators
 - Matrices
 - General fractions without bars
 - Greedy center binary operators
 - Standard center binary operators
- Regular expressions loop replacements
- Regular expressions post-replacements
- Simple post-replacements
- Escapable characters
- Examples

Command line interfaces

1. `sugartex` :

```
Usage: sugartex [OPTIONS] [TO]
```

Reads from stdin and writes to stdout. Can have single argument/option only.

When no args or the arg is not from options then run Pandoc SugarTeX filter that iterates over math blocks.

Options:

```
--kiwi    Same as above but with kiwi flavor,  
--help    Show this message and exit.
```

2. `pre-sugartex` :

Usage: pre-sugartex [OPTIONS]

Reads from stdin and writes to stdout.

When no options: only replace

U+02CE Modifier Letter Low Grave Accent
(that looks like low ``') with \$

Options:

- all Full SugarTeX replace with regexp,
- kiwi Same as above but with kiwi flavor,
- help Show this message and exit.

[Panflute](#) scripts are also installed so you can use it in default Panflute [automation interface in metadata](#) or recommended [panfl CLI](#):

- `panfl sugartex --to markdown ,`
- `panfl sugartex.kiwi -t markdown .`

Examples. Windows:

```
chcp 65001 > NUL
set PYTHONIOENCODING=utf-8

type doc.md | ^
pre-sugartex | ^
pandoc -f markdown --filter sugartex -o doc.md.md
```

Unix:

```
export PYTHONIOENCODING=utf-8

cat doc.md | \
```

```
pre-sugartex | \  
pandoc -f markdown --filter sugartex -o doc.md.md
```

Or splitting Pandoc reader-writer:

```
chcp 65001 > NUL  
set PYTHONIOENCODING=utf-8  
  
type doc.md | ^  
pre-sugartex | ^  
pandoc -f markdown -t json | ^  
sugartex --kiwi | ^  
pandoc -f json -o doc.md.md
```

Tweaking SugarTeX

SugarTeX is written in python and has a tweakable architecture. As you can see in [this filter](#) tweaks can be made in between:

```
sugartex = SugarTeX(ready=False)  
...  
sugartex.ready()
```

Attributes of instance of `SugarTeX` class can be changed. See them in defining of `SugarTeX` class and in it's `__init__` method [here](#). List of attributes:

- `.brackets`
- `.brackets_types`
- `.simple_pre`

- `.superscripts`
- `.subscripts`
- `.regex_pre`
- `.null_ops` (class `NullOps`)
- `.pref_un_ops` (class `PrefUnOps`), including:
 - `.styles` (class `Styles`)
 - `.other_styles` (class `OtherStyles`)
 - `.pref_un_greedy` (class `PrefUnGreedy`)
- `.postf_un_ops` (class `PostfUnOps`)
- `.bin_centr_ops` (class `BinCentrOps`), including:
 - `.matrices` (class `Matrices`)
 - `.bin_centr_greedy` (class `BinCentrGreedy`)
- `.loop_regexps`
- `.regex_post`
- `.simple_post`
- `.escapes`

SugarTeX replacements and operators

Many replacements use `amsmath` macros.

Math delimiters

In default use-case SugarTeX first preprocesses text replacing `,` with `$` (modifier letter low grave accent U+02CE). Can be escaped: `\,`

SugarTeX Completions for Atom:

- `_ ← _``,
- `_ ← \$`.

New escape character

In SugarTeX the default escape character is `\`. But it's a special symbol in LaTeX. In cases when `\` would work as escaping character you can use ``` or ``` (modifier letter grave accent). At the end it will be replaced with `\`.

SugarTeX Completions for Atom:

- `` ← \`` (modifier letter grave accent).

Brackets

Independently replace brackets:

- `.(→ \left({` and `.) → }\right)` (modifier letter low ring U+02F3),
- `..(→ \bigl(` and `).. → \bigr)`,
- `..(→ \Bigl(` and `).. → \Bigr)`,
- `..(→ \biggl(` and `).. → \biggr)`,
- `..(→ \Biggl(` and `).. → \Biggr)` (modifier letter low vertical line U+02CC).

Instead of `(` and `)` can be other brackets:

- `[→ [` and `] →]`,
- `(→ (` and `) →)`,

- `{` → `\{` and `}` → `\}`,
- `|` → `\vert` (box drawings light vertical U+2502, for math in markdown tables),
- `|` → `\vert`,
- `||` → `\Vert` (double vertical line U+2016),
- `<` → `.` and `>` → `.` (modifier letter low left/right arrowhead U+02F1/U+02F2),
- `<` → `\langle` and `>` → `\rangle` (mathematical left/right angle bracket U+27E8/27E9),
- `⌊` → `\lfloor` and `⌋` → `\rfloor` (left/right floor U+230A/U+230B),
- `⌈` → `\lceil` and `⌉` → `\rceil` (left/right ceiling U+2308/U+2309).

SugarTeX Completions for Atom:

Use these shortcuts for fast Unicode typing in Atom:

- `.` ← `\&`,
- `.` ← `_o\small`,
- `.` ← `_'\small`.
- `|` ← `\|`,
- `||` ← `\||`,
- `<` ← `_<`,
- `>` ← `_>`,
- `<>` ← `_<>`,
- `<` ← `\<\`,
- `>` ← `\>\`,
- `<>` ← `\<>\`,

- \lfloor \leftarrow `\lfloor` ,
- \rfloor \leftarrow `\rfloor` ,
- \lceil \leftarrow `\lceil` ,
- \rceil \leftarrow `\rceil` .

Simple pre-replacements

- $\sqrt[3]{}$ \rightarrow `3\sqrt` (cube root U+221B),
- $\sqrt[4]{}$ \rightarrow `4\sqrt` (fourth root U+221C),
- $\,$ \rightarrow `\,` (thin space U+2009).

SugarTeX Completions for Atom:

- $\,$ \leftarrow `\,` (thin space),
- $\,$ \leftarrow `\], [` (thin space),
- $\sqrt{}$ \leftarrow `\^1/2` ,
- $\sqrt[3]{}$ \leftarrow `\^1/3` ,
- $\sqrt[4]{}$ \leftarrow `\^1/4` .

Superscripts and Subscripts

Groups of superscript Unicode characters like 123 are replaced with `^{\{123\}}` . Unless they are escaped with `\` or followed by $\sqrt{}$:

- `\^{123}\sqrt` \rightarrow `^{123}\sqrt` (square root U+221A),
- `\^{123}` \rightarrow `^{\{23\}}` ,
- `^{123} a b c` \rightarrow `^{\{123abc\}}` .

Same is for groups of subscript Unicode characters:

- `_{123}` \rightarrow `_{1-}\{23\}` .

- $\substack{1\,2\,3\,k\,l\,m} \rightarrow _{{123k1m}} .$

List of supported characters can be found in the beginning of the SugarTeX [source code](#).

UPDATE

Now $\langle \rangle$ and $\lceil \rceil$ from [Styles with special brackets](#) end up inside $_{{}} / \wedge{{}}$, like: $A\langle_{ae}\rangle \rightarrow A_{{\langle ae \rangle}}$. Does not work if there are non-subscript/superscript characters inside $\langle \rangle / \lceil \rceil$, like: $A\langle^ae\rangle \rightarrow A\wedge^{{a}}{e}$.

SugarTeX Completions for Atom:

- $\substack{1} \leftarrow \backslash_1 ,$
- $\substack{a} \leftarrow \backslash_a ,$
- $\supset^1 \leftarrow \backslash^1 ,$
- $\supset^a \leftarrow \backslash^a .$

Regular expressions pre-replacements

Nothing. But can be tweaked.

Nullary operators

Big operators replacements:

- $\sum \rightarrow \backslash sum$ (n-ary summation U+2211),
- $\sum: \rightarrow \backslash sum\backslash nolimits ,$
- $\sum: \rightarrow \backslash sum\backslash limits$ (braille pattern dots-48 U+2888).

Supported symbols for limits:

- $\dot{:}$, $\ddot{:}$ \rightarrow `\limits` (braille pattern dots-48/dots-17 U+2888/U+2841),
- $\dot{:}$, $\ddot{:}$, $\ddot{:}$ \rightarrow `\nolimits` (braille pattern dots-23/dots-56 U+2806/U+2830).

Supported big operators:

- Σ \rightarrow `\sum`,
- Π \rightarrow `\prod`,
- \int \rightarrow `\int`,
- \iint \rightarrow `\iint`,
- \iiint \rightarrow `\iiint`,
- \iiint \rightarrow `\iiint`,
- \oint \rightarrow `\oint`.

Who knows what I was thinking about by adding them here instead of Regular expressions replacements...

SugarTeX Completions for Atom:

- $\dot{:}$ \leftarrow `\dot{:}`,
- $\ddot{:}$ \leftarrow `\ddot{:}`, `\small`,
- Σ \leftarrow `\sum`,
- Π \leftarrow `\prod`,
- \int \leftarrow `\int`,
- \iint \leftarrow `\iint`,
- \iiint \leftarrow `\iiint`,
- \iiint \leftarrow `\iiint`,
- \oint \leftarrow `\oint`.

Prefix unary operators

Styles

Text inside standard brackets (`()` , `[]` , `{}`) with special prefix is replaced with style operator. For example:

`[rtext]` or `[{r}text]` \rightarrow `\mathrm{text}` .

First SugarTeX finds opening part like `[{r}` then searches for the first non-escaped closing part `]` that is not inside `{}` or `<>` – SugarTeX counts opening and closing `{<>}` (`<>` would later be replaced with `{}` so both are counted together). For example:

`(rsome{te})(t)` \rightarrow `\mathrm{some{te}(t)}` .

List of available styles:

- `{rtext}` / `{{r}text}` \rightarrow `\mathrm{text}` (**math regular**),
- `{ix}` / `{{i}x}` \rightarrow `\mathit{x}` (**math italic**),
- `{bx}` / `{{b}x}` \rightarrow `\mathbf{x}` (**math bold**),
- `{\betax}` / `{{\beta}x}` \rightarrow `\boldsymbol{x}` (**math bold italic**),
- `{mtext}` / `{{m}text}` \rightarrow `\mathtt{text}` (**math monospace**),
- `{cA}` / `{{c}A}` \rightarrow `\mathcal{A}` (**math calligraphic**, no cyrillic support, see Monotype Corsiva),
- `{ttext}` / `{{t}text}` \rightarrow `\text{text}` (**text**),
- `{t itext}` / `{{t i}text}` \rightarrow `\textit{text}` (**text italic**),
- `{t btext}` / `{{t b}text}` \rightarrow `\textbf{text}` (**text bold**),
- `{t \betatext}` / `{{t \beta}text}` \rightarrow `\textit{\textbf{text}}` (**text bold italic**),

- $\{\vec{x}\}$ / $\{\mathbf{x}\}$ \rightarrow `\mathbf{x}` (**vector bold notation**, combining right arrow above U+20D7, first one is ‘space’ + $\vec{}$),
- $\{\dot{x}\}$ / $\{\ddot{x}\}$ \rightarrow `\mathbf{x}` (**vector bold notation**, braille pattern dots-45/dots-12 U+2818/U+2803 [right upper 2/left upper 2]),
- $\{\dot{A}\}$ / $\{\ddot{A}\}$ \rightarrow `\mathbf{A}` (**matrix bold notation**, braille pattern dots-124/dots-1245 U+280B/U+281B).

SugarTeX Completions for Atom:

- $\vec{}$ \leftarrow `\^->`,
- $\dot{}$ \leftarrow `\^:`,
- $\ddot{}$ \leftarrow `\^::`,
- $\ddot{}$ \leftarrow `\array`,
- $\ddot{}$ \leftarrow `\^:.\rot-90`,
- $\ddot{}$ \leftarrow `\matrix`.

Styles with special brackets

- $\langle^{\beta}\text{text}\rangle$ / $\langle^{\beta}\{\text{text}\}\rangle$ \rightarrow `\textit{\textbf{text}}` (**text bold italic**),
- $\langle^i\text{text}\rangle$ / $\langle^i\{\text{text}\}\rangle$ \rightarrow `\textit{text}` (**text italic**),
- $\langle^b\text{text}\rangle$ / $\langle^b\{\text{text}\}\rangle$ \rightarrow `\textbf{text}` (**text bold**),
- $\langle\text{text}\rangle$ \rightarrow `\text{text}` (**text regular**, single left/right-pointing angle quotation mark U+2039/U+203A),
- ‘text’ \rightarrow `\mathrm{text}` (**math regular**, modifier letter begin/end high tone U+02F9/U+02FA).

SugarTeX Completions for Atom:

- \langle \leftarrow `\<`,

- `>` \leftarrow `\>`,
- `<>` \leftarrow `\<>`,
- `<>` \leftarrow `\text`,
- `{}^r` \leftarrow `\^r\small`,
- `{}^r` \leftarrow `\regular`.

Greedy prefix unary operators

- `{\in smth}` / `{\in smth}_` \rightarrow `\begin{cases} smth\end{cases}`
(**piecewise**, element of with long horizontal stroke U+22F2).

```
``
  .|x|. = {\in x. <if> x\ge 0 |
          -x. <if> x<0 }
``
```

SugarTeX finds non-escaped `{\in` or `{\in_` first then searches for non-escaped `}` or `_` that is not inside `{}` or `<>` – SugarTeX counts opening and closing `{\in_<>` (`<>` would later be replaced with `{}` so both are counted together).

SugarTeX Completions for Atom:

- `\in` \leftarrow `\-e`,
- `\in` \leftarrow `\-E`.

Standard prefix unary operators

- `<matrix a` \rightarrow `\begin{matrix} a`
(left-pointing curved angle bracket U+29FC),

- $A^2 a \rightarrow \text{\vphantom{A}^2} a$
(invisible characters that adjust height, ghost U+1F47B),
- $\rightarrow^{\text{text}} a \rightarrow \text{\xrightarrow{text}} a$
(arrow with text above that adjusts to the text length, rightwards arrow U+2192, top square bracket U+23B4),
- $\leftarrow^{\text{long text}} a \rightarrow \text{\xleftarrow{{long text}}} a$
(leftwards arrow U+2190).

SugarTeX finds non-escaped `< *` first (for example) then searches for a place before non-escaped `}`, `,`, space, newline or end of the string that is not inside `{}` or `<>` – SugarTeX counts opening and closing `{}<>` (`<>` would later be replaced with `{}` so both are counted together).

SugarTeX Completions for Atom:

- `< ← \<\alt2 ,`
- `> ← \>\alt2 ,`
- `<> ← \<>\alt2 ,`
- \ghost ← `\ghost ,`
- \rightharpoonup ← `\^^ ,`
- \rot90 ← `\^]\rot90 ,`
- \rightarrow ← `\-> ,`
- \leftarrow ← `\<- .`

Postfix unary operators

- $a \vec{x} \rightarrow a \text{\vec{x}}$ (vector, combining right arrow above U+20D7),

- `a x^{\rightarrow}` → `a \overrightarrow{x}` (**arrow above**, combining right harpoon above U+20D1),
- `a x^{\widehat{}}` → `a \widehat{x}` **warning:** works only if the next character after `^` is `}`, `,`, `>`, newline or end of the string,
- `a x^{\circ}` → `a \hat{x}` (modifier letter circumflex accent U+02C6),
- `a x^{\bar{}}` → `a \bar{x}` (macron U+00AF),
- `a x^{\overline{}}` → `a \overline{x}` (overline U+203E),
- `a x^{\cdot}` → `a \dot{x}` (dot above U+02D9),
- `a x^{\ddot{}}` → `a \ddot{x}` (diaeresis U+00A8),
- `x + y + z^{\overbrace{}}` → `x + \overbrace{y + z}` (top curly bracket U+23DE),
- `x + {y + z}^{\underbrace{}}` → `x + \underbrace{\{y + z\}}` (bottom curly bracket U+23DF),
- `a x_{\underline{}}` → `a \underline{x}` **warning:** works only if the next character after `_` is `}`, `,`, `>`, newline or end of the string (modifier letter low macron U+02CD),
- `a matrix\end{}` → `a \end{matrix}` (right-pointing curved angle bracket U+29FD),

SugarTeX finds non-escaped `*\end{}` first (for example) then before it searches for a place after non-escaped `{`, `<`, space, newline or start of the string that is not inside `{}` or `<>` – SugarTeX counts opening and closing `{}`, `<>` (`<>` would later be replaced with `{}` so both are counted together).

In combination with styles:

When combining **one-character** postfix unary operators with styles the order in which operators are applied changes:

`[bx→]` \rightarrow `\vec{\mathbf{x}}`

SugarTeX Completions for Atom:

- `→` \leftarrow `\^->`,
- `↗` \leftarrow `\^->\har`,
- `^` \leftarrow `\^{\small}`,
- `ˆ` \leftarrow `\^_{\small}` (macron),
- `˘` \leftarrow `\^-\small` (macron),
- `¯` \leftarrow `\^_` (overline),
- `·` \leftarrow `\^.`,
- `¨` \leftarrow `\^..`,
- `↻` \leftarrow `\^}\rot90`,
- `↺` \leftarrow `_}\rot-90`,
- `_` \leftarrow `_`,
- `<` \leftarrow `\<\alt2`,
- `>` \leftarrow `\>\alt2`,
- `<>` \leftarrow `\<>\alt2`.

Center binary operators

Matrices

Family of `*matrix` amsmath macros is given by `|:` operator (broken bar U+00A6, braille pattern dots-124 U+280B):

`[a .b |: c .d]` \rightarrow
`\begin{bmatrix}a .b|c .d\end{bmatrix}` \rightarrow
`\begin{bmatrix}a &b\\c &d\end{bmatrix}`

All brackets:

- $\langle a \ .b \ | \ddot{c} \ .d \rangle \rightarrow \dots\text{matrix}\dots$ (**no brackets**,
modifier letter low left/right arrowhead U+02F1/U+02F2),
- $\{a \ .b \ | \ddot{c} \ .d\} \rightarrow \dots\text{Bmatrix}\dots$ (**curly brackets**),
- $\langle (a \ .b \ | \ddot{c} \ .d) \rangle / \{(a \ .b \ | \ddot{c} \ .d)\} \rightarrow \dots\text{pmatrix}\dots,$
- $\langle [a \ .b \ | \ddot{c} \ .d] \rangle / \{[a \ .b \ | \ddot{c} \ .d]\} \rightarrow \dots\text{bmatrix}\dots,$
- $\langle |a \ .b \ | \ddot{c} \ .d| \rangle / \{|a \ .b \ | \ddot{c} \ .d|\} /$
 $\langle |a \ .b \ | \ddot{c} \ .d| \rangle / \{|a \ .b \ | \ddot{c} \ .d|\} \rightarrow \dots\text{vmatrix}\dots$
 (box drawings light vertical U+2502, for math in markdown tables),
- $\langle ||a \ .b \ | \ddot{c} \ .d|| \rangle / \{||a \ .b \ | \ddot{c} \ .d||\} \rightarrow \dots\text{Vmatrix}\dots$
 (double vertical line U+2016).

SugarTeX finds non-escaped binary operator separator $| \ddot{}$ first then:

- searches for a place after non-escaped $\{$ or \langle that is not inside $\{\}$ or $\langle \rangle$,
- searches for a place before non-escaped $\}$ or \rangle that is not inside $\{\}$ or $\langle \rangle$,
- it also figures out bracket type properly,
- this way it finds two arguments (SugarTeX counts opening and closing $\{\}\langle \rangle$, $\langle \rangle$ would later be replaced with $\{\}$ so both are counted together).

SugarTeX Completions for Atom:

- $\cdot \leftarrow \backslash \&$,
- $\cdot \leftarrow \backslash _o \backslash \text{small}$,
- $| \leftarrow \backslash |$,
- $|| \leftarrow \backslash ||$,

- \leftarrow `_<`,
- \rightarrow `_>`,
- \leftrightarrow `_<>`,
- $\|$ `\|`,
- $\|$ `\--\rot90`,
- $\ddot{}$ `\^:.\rot-90`,
- $\ddot{}$ `\matrix`.

General fractions without bars

Fractions works almost the same as Matrices - they add brackets and stack arguments: first arg is atop of the second arg. But with some differences:

- they use $\dfrac{}{}$ or $\frac{}{}$ as a separator (broken bar U+00A6, braille pattern dots-45 U+2818 / dots-12 U+2803),
- cannot handle more than one line break (so two args only),
- they use `\genfrac` `amsmath` macro,
- they can have size modifiers after $\dfrac{}{}$:
 - `d / ^{d}` - display mode,
 - `t / ^{t}` - text mode,
 - `s / ^{s}` - smaller,
 - `xs / ^{xs}` - extra small,
- left and right brackets can be different.

Examples:

- $\left(x \dfrac{}{} y \right)$,
- $\left[x \dfrac{}{} y \right]$,
- $\{ x \dfrac{}{} y \}$ (**curly brackets**),

- $\langle x | : y \rangle$ (**no brackets**, modifier letter low left/right arrowhead U+02F1/U+02F2),
- $\langle | x | : y | \rangle$, $\langle | x | : y | \rangle$ (box drawings light vertical U+2502, for math in markdown tables),
- $\langle || x | : ^d y || \rangle$ (double vertical line U+2016).

Arguments search algorithm is the same as for matrices.

SugarTeX Completions for Atom:

- $| \leftarrow \backslash |$,
- $|| \leftarrow \backslash ||$,
- $\langle \leftarrow \backslash _\langle$,
- $\rangle \leftarrow \backslash _\rangle$,
- $\langle \rangle \leftarrow \backslash _\langle \rangle$,
- $| \leftarrow \backslash \backslash$,
- $| \leftarrow \backslash -- \backslash \text{rot}90$,
- $: \leftarrow \backslash ^{:}$.

Greedy center binary operators

Arguments search algorithm is the same as for matrices (except it now does not have brackets).

1. $\langle \text{smth1} | ::^t \text{smth2} \rangle \rightarrow$
 $\backslash \text{begin}\{\text{smallmatrix}\}\text{smth1}|\text{smth2}\backslash \text{end}\{\text{smallmatrix}\}$,
 (Braille Pattern Dots-1245 U+281B).

$\dots (\langle a \cdot b | ::^t c \cdot d \rangle) \dots$

2. $\langle \text{smth1} \mid \text{smth2} \rangle \rightarrow$
 $\backslash \text{begin}\{\text{array}\}\text{smth1}\mid\text{smth2}\backslash \text{end}\{\text{array}\},$
 (Braille Pattern Dots-1245 U+281B).

```

..
. [ <                                     < cccc | c >
    x11 . x12 . x13 . ... . x1n | ⋮
    x21 . x22 . x23 . ... . x2n |
    ⋮ . ⋮ . ⋮ . ⋮ . ⋮ |
    xp1 . xp2 . xp3 . ... . xpn > ] .
..

```

3. $\langle \text{smth1} \mid \# \text{smth2} \rangle \rightarrow$
 $\backslash \text{begin}\{\text{aligned}\}\text{smth1}\mid\text{smth2}\backslash \text{end}\{\text{aligned}\},$

```

..
. | x | . = . { < x . < if > x ≥ 0 | #
                - x . < if > x < 0 > > .
..

```

4. $\langle \text{smth1} \mid _ \text{smth2} \rangle / \langle \text{smth1} \mid _ \text{smth2} \rangle \rightarrow$
 $\backslash \text{substack}\{\text{smth1}\mid\text{smth2}\},$
 (modifier letter shelf U+02FD / bottom square bracket U+23B5)

```

.. Σ0 ≤ i ≤ N | _ 0 ≤ j ≤ M (ij)3 ..

```

5. $\langle \text{smth1} \mid _{}^1 \text{smth2} \rangle / \langle \text{smth1} \mid _{}^1 \text{smth2} \rangle \rightarrow$
 $\backslash \text{begin}\{\text{subarray}\}\{1\}\text{smth1}\mid\text{smth2}\backslash \text{end}\{\text{subarray}\},$
 (modifier letter shelf U+02FD / bottom square bracket U+23B5)

```

.. Σ0 ≤ i ≤ N | _{}^1 0 ≤ j ≤ M (ij)3 ..

```

Instead of `l` (left) it can also be `c` (center) or `r` (right).

SugarTeX Completions for Atom:

- `::` \leftarrow `\^::` ,
- `_` \leftarrow `_` ,
- `_` \leftarrow `_]\rot-90` ,
- `_` \leftarrow `_]\rot-90\alt` ,
- `|` \leftarrow `\|` ,
- `|` \leftarrow `\--\rot90` .

Standard center binary operators

Fractions

- `x/y` \rightarrow `\frac{x}{y}` (division slash U+2215),
- `1+x/y` \rightarrow `\frac{1+x}{y}` ,
- `1 + {x + z}/y` \rightarrow `1 + \frac{{x + z}}{y}` ,
- `x/dy` \rightarrow `\dfrac{x}{y}` ,
- `x/ty` \rightarrow `\tfrac{x}{y}` ,
- `x/cy` \rightarrow `\cfrac{x}{y}` ,
- `x/sy` and `x/xsy` are the same as `x/ty` but smaller and use `\genfrac` macros. Bar thickness can be set this way: `{0.5px}x/sy` .

Roots, overset, underset

- `√64` \rightarrow `\sqrt[]{64}` (square root U+221A),
- `6√64` \rightarrow `\sqrt[6]{64}` ,
- `1 + 6√64` \rightarrow `1 + \sqrt[6]{64}` ,

- `'lim'_{x\rightarrow 0}` / `'lim'_{x\rightarrow 0} \rightarrow \underset{x\rightarrow 0}{\{'lim'\}}` (modifier letter shelf U+02FD / bottom square bracket U+23B5),
- `{x + ... + x}^{\sim}\{k \langle times \rangle\}` \rightarrow `\overset{\{k \langle times \rangle\}}{\{x + ... + x\}^{\sim}}` (top square bracket U+23B4).

Binomial coefficients

- $(i|{}^c n) \rightarrow \backslash\text{binom}\{i\}\{n\}$,
- $(i|{}^c d n) \rightarrow \backslash\text{dbinom}\{i\}\{n\}$ (display),
- $(i|{}^c t n) \rightarrow \backslash\text{tbinom}\{i\}\{n\}$ (text).

In this case SugarTeX finds non-escaped binary operator separator `|` first then searches for `(` and `)`. Other stop symbols do not work.

SugarTeX finds non-escaped binary operator separator (like `/`) first then:

- searches for a place after non-escaped `{`, `<`, space, newline or start of the string that is not inside `{}` or `<>`,
- searches for a place before non-escaped `}`, `>`, space, newline or end of the string that is not inside `{}` or `<>`,
- this way it finds two arguments (SugarTeX counts opening and closing `{}`, `<>` would later be replaced with `{}` so both are counted together).

SugarTeX Completions for Atom:

- `␣` ← `_`,
- `␣` ← `_]\rot-90`,
- `␣` ← `_]\rot-90\alt`,
- `␣` ← `\^^`,

- $\lrcorner \leftarrow \text{\textbackslash}\wedge\text{\textbackslash}\text{rot90}$,
- $/ \leftarrow \text{\textbackslash}/$,
- $\sqrt{} \leftarrow \text{\textbackslash}\wedge 1/2$,
- $| \leftarrow \text{\textbackslash}|$,
- $\!| \leftarrow \text{\textbackslash}\text{--}\text{\textbackslash}\text{rot90}$.

Regular expressions loop replacements

Nothing. But can be tweaked.

Regular expressions post-replacements

Nothing. But can be tweaked.

Simple post-replacements

- $\!| \rightarrow \text{\textbackslash}|$ (broken bar U+00A6, this should be after other $|$ replacements),
- $\cdot \rightarrow \&$ (modifier letter low ring U+02F3, this should be after brackets and other \cdot replacements),
- $\leftarrow \rightarrow \{$ and $\rightarrow \rightarrow \}$ (modifier letter low left/right arrowhead U+02F1/U+02F2),
- $_ \rightarrow _$ (modifier letter low macron U+02CD),
- $\` \rightarrow \text{\textbackslash}$,
- $\` \rightarrow \text{\textbackslash}$ (modifier letter grave accent U+02CB),
- $\updownarrow\{d\} \rightarrow \text{\textbackslash}\text{displaystyle}$ (up down arrow U+2195),
- $\updownarrow\{t\} \rightarrow \text{\textbackslash}\text{textstyle}$,
- $\updownarrow\{s\} \rightarrow \text{\textbackslash}\text{scriptstyle}$,

- $\uparrow^{\{xs\}}$ \rightarrow `\scriptscriptstyle` ,
- Superscripts and Subscripts replacements give:
- \uparrow^d \rightarrow `\displaystyle` ,
- \uparrow^t \rightarrow `\textstyle` ,
- \uparrow^s \rightarrow `\scriptstyle` ,
- \uparrow^{xs} \rightarrow `\scriptscriptstyle` .

SugarTeX Completions for Atom:

- `|` \leftarrow `\\` ,
- `|` \leftarrow `\--\rot90` ,
- `.` \leftarrow `\&` ,
- `.` \leftarrow `_o\small` ,
- `<` \leftarrow `_<` ,
- `>` \leftarrow `_>` ,
- `<>` \leftarrow `_<>` ,
- `_` \leftarrow `_` ,
- ``` \leftarrow `\`` (modifier letter grave accent).
- \updownarrow \leftarrow `\<->\rot90` .

Escapable characters

All one-character replacements from:

- Prefix unary operators,
- Postfix unary operators,
- Center binary operators,
- Nullary operators,
- Simple pre-replacements,

- Simple post-replacements,

and €, ›, ¸, ⇕, ¸.

(element of with long horizontal stroke U+22F2, single right-pointing angle quotation mark U+203A, modifier letter end high tone U+02FA, up down arrow U+2195, modifier letter low vertical line U+02CC)

are escapable with \.

Examples

You can find SugarTeX examples [in this document](#) (SugarTeX code + rendered formulas).