

OneHotkey -- Math Formula Input Simplification Tool

This is a script that simplifies math formula inputs in `oneNote` and `word` with `AutoHotKey` script, e.g., `\a` for α (`\alpha`).

Demonstration video (Early version):

- [AutoHotKey增强OneNote公式输入测试](#)哔哩哔哩 bilibili

This project is still updating. Your suggestions and contributions are welcome.

If the formulas aren't displayed correctly, go to [README EN.pdf](#).

How to Use

1. Download and run [OneHotkey.exe](#).
2. Input the code of the symbol, then press `Space` to get the symbol. For example, input `\a` and press `Space` to get α .
3. For editing the symbol mapping, please refer to [Code Editing Guide](#). If you need help, go to the [AutoHotKey official website](#).
4. To stop the script, right click the `H` icon in the system tray and select `Exit`.

Table of Contents

- [OneHotkey -- Math Formula Input Simplification Tool](#)
 - [How to Use](#)
 - [Table of Contents](#)
 - [Symbol Mapping](#)
 - [Overview](#)
 - [Full Table](#)
 - [Frequently Used Letters](#)
 - [Operators](#)
 - [Greek Letters](#)
 - [Matrix](#)
 - [Modifiers](#)
 - [Arrows](#)
 - [Symbols](#)
 - [Structures](#)
 - [Prefix for Fancy Letters](#)
 - [Multi-column Equations](#)
 - [Theorem Structures](#)

- o [Recommendations](#)
- o [Experimental Features \(In folder `experimental/`\)](#)
- o [Code Editing Guide](#)

Symbol Mapping

Overview

The script contains multiple symbol mappings, including Greek letters, math fonts, frequently used letters, and structures. The following is a list of some typical mappings. Make sure that you have entered the formula input mode with `Alt + =`.

Code	Output	Category	Source
<code>\a</code>	α	lowercase Greek letters	<code>\alpha</code>
<code>\D</code>	Δ	uppercase Greek letters	<code>\Delta</code>
<code>\R</code> , <code>\C</code> , <code>\Z</code> , <code>\N</code> , <code>\J</code>	\mathbb{R} , \mathbb{C} , \mathbb{Z} , \mathbb{N} , \mathbb{J}	frequently used letters	<code>\doubleR</code> , ..., <code>\doubleN</code>
<code>\do x</code> , <code>\sc x</code> , <code>\fr x</code>	\mathbb{X} , \mathcal{X} , \mathfrak{X}	fancy letter forms	<code>\doubleX</code> , <code>\scriptX</code> , <code>\frakturX</code>
<code>\m3</code> , <code>\m4</code> , ...	specific shape matrices	matrices	<code>[\matrix(@@@)]</code> , ...
<code>x\h</code> , <code>x\~</code> , <code>x\d2</code>	\hat{x} , \tilde{x} , \ddot{x}	modifiers	<code>\hat</code> , <code>\tilde</code> , <code>\ddot</code>
<code>\x</code> , <code>\X</code> , <code>\sq</code> , <code>\pa</code> , <code>\eq</code>	\cdot , \times , $\sqrt{\square}$, \parallel , \equiv	operators	<code>\cdot</code> , <code>\times</code> , <code>\sqrt</code> , <code>\parallel</code> , <code>\equiv</code>
<code>\pd</code> , <code>\d</code> , <code>\dt</code> , <code>\inf</code>	∂ , d , $\frac{d}{dt}$, ∞	frequently used symbols	<code>\partial</code> , <code>d</code> , <code>d/dt</code> , <code>\infty</code>
<code>\limx</code> , <code>\limx0</code>	$\lim_{x \rightarrow \infty}$, $\lim_{x \rightarrow 0}$	limits	<code>lim_(x->\infty)</code> , <code>lim_(x->0)</code>
<code>\ls</code>	\mathring{P}	left super-and-lowerscript	<code>_P</code>
<code>\i</code> , <code>\j</code> , <code>\k</code>	i, j, k	imaginary/quaternion symbols	<code>"i"</code> , <code>"j"</code> , <code>"k"</code>
<code>\ejw</code>	$e^{j\omega}$	complex exponential factor	<code>e^j\omega</code>

You shall notice that `(space)` is commonly used, which is the key feature of OneNote formula input.

Full Table

Frequently Used Letters

Code	Output	Source	Code	Output	Source
\pd	∂	\partial{Space}	\d	d	"d"{Space}
\inf	∞	\infty{Space}	\dt	$\frac{d}{dt}$	"d"{Space}/"d"{Space}t{Space}
\R	\mathbb{R}	\doubleR{Space}	\E	$\mathbb{E}[\square]$	\doubleE{Space}[]{Space}{Left}
\Q	\mathbb{Q}	\doubleQ{Space}	\z	\mathbb{Z}	\doublez{Space}
\N	\mathbb{N}	\doubleN{Space}	\c	\mathbb{C}	\doublec{Space}
\J	\mathbb{J}	\doubleJ{Space}	\n	∇	\nabla{Space}

Operators

Code	Output	Source	Code	Output	Source
\x	.	\cdot{Space}	\x	\times	\times{Space}
\sq	$\sqrt{\square}$	\sqrt{Space 2}{Left}	\pa	\parallel	\parallel{Space}
\ss	\subset	\subset{Space}	\sse	\subseteq	\subsetneq{Space}
\op	\oplus	\oplus{Space}	\ox	\otimes	\otimes{Space}
\od	\odot	\odot{Space}	\dd	\ddots	\ddots{Space}
\cd	\cdots	\cdots{Space}	\vd	\vdots	\vdots{Space}
\map	\mapsto	\mapsto{Space}	\pro	\propto	\propto{Space}
\as	\because	\because{Space}	\so	\therefore	\therefore{Space}
\eq	\equiv	\equiv{Space}	\deq	\triangleq	\triangleq{Space}
\xe	$\times 10^{\square}$	\times{Space}10^{\square}{Space}{Left}	\ex	\exists	\exists{Space}
\fa	\forall	\forall{Space}	\ppd	$\frac{\partial}{\partial}$	\partial{Space}/\partial{Space 2}{Left 3}
\ppd	$\frac{\partial}{\partial}$	\partial{Space}/\partial{Space 2}{Left 3}			

Greek Letters

Code	Output	Source	Code	Output	Source
\a	α	\alpha{Space}	\b	β	\beta{Space}
\e	ε	\varepsilon{Space}	\ve	ϵ	\epsilon{Space}
\de	δ	\delta{Space}	\D	Δ	\Delta{Space}
\s	σ	\sigma{Space}	\s	Σ	\Sigma{Space}
\l	λ	\lambda{Space}	\L	Λ	\Lambda{Space}

Code	Output	Source	Code	Output	Source
\t	θ	\theta{space}	\T	Θ	\Theta{space}
\p	ϕ	\phi{space}	\P	Φ	\Phi{space}
\o	ω	\omega{space}	\O	Ω	\Omega{space}
\g	γ	\gamma{space}	\G	Γ	\Gamma{space}

- \ve means variant epsilon. For convenience, \e is set to ε and \ve is set to ϵ , which is different from their original code.

Matrix

Code	Output	Source
\m4	4 by 4 empty matrix	[\matrix(@@@&&){Space}]{Space}
\m3	3 by 3 empty matrix	[\matrix(@@&&){Space}]{Space}
\m2	2 by 2 empty matrix	[\matrix(@&){Space}]{Space}
\m	empty matrix awaiting & @ to set size	[]{Space}{Left}\matrix(){Left}

Modifiers

Code	Output	Source
\d1	\dot{x}	\dot{Space} 2
\d2	\ddot{x}	\ddot{Space} 2
\d3	3 dots above	\dddot{Space} 2
\d4	4 dots above	\dddot{Space} 2
\~	\tilde{x}	\tilde{Space} 2
\v	\vec{x}	\vec{Space} 2
\h	\hat{x}	\hat{Space} 2
\ub	\underline{x}	\underbar{Space} 2{Left}

- For the above codes, you should input like x\h.

Arrows

Code	Output	Source	Code	Output	Source
\lr	\leftrightarrow	\leftrightarrow{Space}	\Lr	\Leftrightarrow	\Leftrightarrow{Space}
\lrs	$\Leftarrow\Rightarrow$	\leftrightsquigarrow{Enter}{Left}	\la	\leftarrow	\leftarrow{Space}

Code	Output	Source	Code	Output	Source
\La	\Leftarrow	\Leftarrow{Space}	\ra	\rightarrow	\rightarrow{Space}
\Ra	\Rightarrow	\Rightarrow{Space}	\down	\downarrow	\downarrow{Space}
\up	\uparrow	\uparrow{Space}			

Symbols

Code	Output	Source	Code	Output	Source
\deg	$^\circ$	\degree{Space}	\st	$*$	\star{Space}

Structures

Code	Output	Source
\r	{□	\right.{Left}
\leb	□}	\left\box{Space 2}{Left}
\ceil	[□]	\lceil{Space}\rceil{Space 2}{Left}
\floor	[□]	\lfloor{Space}\rfloor{Space 2}{Left}
\brak	(□)	\bra{Space}\ket{Space 2}{Left}
\ls	□P	_ P {Left 4}
\ab	□	\above{Space 2}{Left}
\be	x	\below{Space 2}{Left}
\abb	□x	\overbrace{Space 2}
\beb	□x	\underbrace{Space 2}
\fu	myfunction□	\funcapply
\Norm	□	\norm{Space}\norm{Space 2}{Left}
\limx, \limx0	$\lim_{x \rightarrow \infty}, \lim_{x \rightarrow 0}$	\lim_(x->\infty){Space}, \lim_(x->0{Space}){Space}
\limt, \limt0	$\lim_{t \rightarrow \infty}, \lim_{t \rightarrow 0}$	\lim_(t->\infty){Space}, \lim_(t->0{Space}){Space}
\limn, \limk	$\lim_{n \rightarrow \infty}, \lim_{k \rightarrow \infty}$	\lim_(n->\infty){Space}, \lim_(k->\infty){Space}
\limh	$\lim_{h \rightarrow 0}$	\lim_(h->0{Space}){Space}

Code	Output	Source
\BO		\boxed{Enter}{Left 2}
\qu	Quad space	\quad{Enter}{Left}
\diverge	$\frac{\partial}{\partial x} + \frac{\partial}{\partial y} + \frac{\partial}{\partial z}$	Omitted
\gradient	$\frac{\partial}{\partial x} \vec{a}_x + \frac{\partial}{\partial y} \vec{a}_y + \frac{\partial}{\partial z} \vec{a}_z$	Omitted
\curl	Curl matrix	

- \funcapply is a little different from \of. Have a try by yourself!

Prefix for Fancy Letters

Code	Output	Source
\sc	\mathcal{X}	\script
\do	\mathbb{X}	\double
\fr	\mathfrak{X}	\fraktur

- For these mappings, your input should be like \sc X.

Multi-column Equations

Code	Output	Source
\eq2	Two-column equation	\eqarray(&=@&){Space}{Left 6}
\eqs	Example equation array	\eqarray(a\quad&b@c\quad&e){Enter}{Left 11}

Note: Multi-column equations are used for aligning multiple equations, using @ as placeholder and & as alignment point.

Theorem Structures

Code	Output	Source
\pf	Proof structure	\bProof.\b{Enter 2}!=\{Space}\#\qed\b{Enter}{Left 7}
\thm	Theorem structure	\bTheorem{Space}.\b^i{Space}{Left 2}
\que	Question structure	\bQuestion{Space}.\b{Space}{Left 2}

Recommendations

- Learn more about the math input from this document: [UTN28-PlainTextMath-v3.pdf](#). Page 39~47 is useful.
- Input Unicode characters directly: https://github.com/gtj1/symbol_assist
- Intuitive Vim-like text cursor control: <https://github.com/RUSRUSHB/AutoTextCursor>

Experimental Features (In folder `experimental/`)

`key_combination.exe`

- Use key combinations to input special characters and structures
- Contains: Start formula inputting; Division line; Boxed text; Text block

`rus_hotkey.exe`

- Input Russian alphabets. They can be integrated into formula inputting.
- Format: `\+Romanized Alphabet+R`
- e.g., `\dR` generates Δ , `\DR` generates Δ

Code Editing Guide

For editing the mapping, please: Edit `OneHotkey.ahk`, compile it with `Ahk2Exe`, and run the compiled `.exe` file. You are recommended to learn more about `AutoHotKey` from its [website](#).

The code of `OneHotkey.ahk` is very easy to understand, even if you have not learnt about `AutoHotKey`. For newcomers, the explanation of the code is as follows:

Each line of the code is a mapping of the input code to the output symbol. The format is :

`(parameters):input::output`. For example, `::\a::\alpha` means that when you input `\a`, the script will output `\alpha`.

The script uses global hotstring settings `#Hotstring c o ?`, which apply to all mappings:

Parameter	Meaning
<code>c</code>	Case-sensitive. <code>\a</code> and <code>\A</code> are different.
<code>o</code>	Delete the space you entered at the end.
<code>?</code>	Output formula even if you have typed something before the code. Otherwise, it will fail in cases like <code>x\h</code>