

# OneHotkey -- Math Formula Input Simplification Tool

---

This is a script that simplifies math formula inputs in `OneNote`, `Word` and `PowerPoint` with `AutoHotKey` script, e.g., `\a` for  $\alpha$  (`\alpha`).

这是一个用于简化 `OneNote`, `Word` 和 `PowerPoint` 中数学公式输入的 `AutoHotKey` 脚本, 例如, `\a` 代表 $\alpha$  (`\alpha`).

中文版帮助文档请见[README\\_CN.md](#).

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

## Table of Contents

---

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

## How to Use

---

1. Download and run `symbol_assist.exe`.
2. Input the code of the symbol, then press `Space` to get the symbol. For example, input `\a` and press `Space` to get  $\alpha$ .
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`.

# 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>	$\alpha$	lowercase Greek letters	<code>\alpha</code>
<code>\D</code>	$\Delta$	uppercase Greek letters	<code>\Delta</code>
<code>\R, \C, \Z, \N</code>	$\mathbb{R}, \mathbb{C}, \mathbb{Z}, \mathbb{N}$	frequently used letters	<code>\doubleR, ...</code>
<code>\do x, \sc x, \fr x</code>	$\mathbb{X}, \mathcal{X}, \mathfrak{X}$	fancy letter forms	<code>\doubleX, \scriptx, \frakturx</code>
<code>\m3, \m4, ...</code>	3 by 3 empty matrix, ...	matrices	<code>[\matrix{@@&amp;&amp;}], ...</code>
<code>x\h, x\~, x\d2</code>	$\hat{x}, \tilde{x}, \ddot{x}$	modifiers	<code>\hat, \tilde, \ddot</code>
<code>\x, \X, \sq, \pa, \eq</code>	$\cdot, \times, \sqrt{\square}, \parallel, \equiv$	operators	<code>\cdot, \times, \sqrt, \parallel, \equiv</code>
<code>\pd, \di, \inf</code>	$\partial, d, \infty$	frequently used symbols	<code>\partial, "d", \infty</code>
<code>\ls</code>	$\square P$	left super-and-lowerscript	<code>^_ P</code>
<code>\i, \j, \k</code>	$i, j, k$	imaginary/quaternion symbols	<code>"i", "j", "k"</code>

You shall notice that (space) is commonly used, which is the key feature of OneNote formula input. Capital letter code should be inputted with `Shift`, not `CapsLock`.

## Full Table

### Frequently Used Letters

Code	Output	Source	Code	Output	Source
<code>\pd</code>	$\partial$	<code>\partial{Space}</code>	<code>\di</code>	$d$	<code>"d"</code>
<code>\inf</code>	$\infty$	<code>\infty{Space}</code>	<code>\Z</code>	$\mathbb{Z}$	<code>\doubleZ{Space}</code>
<code>\Q</code>	$\mathbb{Q}$	<code>\doubleQ{Space}</code>	<code>\R</code>	$\mathbb{R}$	<code>\doubleR{Space}</code>

Code	Output	Source	Code	Output	Source
<code>\N</code>	$\mathbb{N}$	<code>\doubleN{Space}</code>	<code>\C</code>	$\mathbb{C}$	<code>\doubleC{Space}</code>
<code>\J</code>	$\mathbb{J}$	<code>\doubleJ{Space}</code>			

## Operators

Code	Output	Source	Code	Output	Source
<code>\x</code>	$\cdot$	<code>\cdot{Space}</code>	<code>\X</code>	$\times$	<code>\times{Space}</code>
<code>\sq</code>	$\sqrt{\square}$	<code>\sqrt{Space 2}{Left}</code>	<code>\pa</code>	$\parallel$	<code>\parallel{Space}</code>
<code>\ss</code>	$\subset$	<code>\subset{Space}</code>	<code>\sse</code>	$\subseteq$	<code>\subseteq{Space}</code>
<code>\op</code>	$\oplus$	<code>\oplus{Space}</code>	<code>\ox</code>	$\otimes$	<code>\otimes{Space}</code>
<code>\od</code>	$\odot$	<code>\odot{Space}</code>	<code>\dd</code>	$\ddots$	<code>\ddots{Space}</code>
<code>\cd</code>	$\cdots$	<code>\cdots{Space}</code>	<code>\vd</code>	$\vdots$	<code>\vdots{Space}</code>
<code>\map</code>	$\mapsto$	<code>\mapsto{Space}</code>	<code>\pro</code>	$\propto$	<code>\propto{Space}</code>
<code>\as</code>	$\because$	<code>\because{Space}</code>	<code>\so</code>	$\therefore$	<code>\therefore{Space}</code>
<code>\eq</code>	$\equiv$	<code>\equiv</code>			

## Greek Letters

Code	Output	Source	Code	Output	Source
<code>\a</code>	$\alpha$	<code>\alpha{Space}</code>	<code>\b</code>	$\beta$	<code>\beta{Space}</code>
<code>\e</code>	$\varepsilon$	<code>\varepsilon{Space}</code>	<code>\ve</code>	$\epsilon$	<code>\epsilon{Space}</code>
<code>\d</code>	$\delta$	<code>\delta{Space}</code>	<code>\D</code>	$\Delta$	<code>\Delta{Space}</code>
<code>\s</code>	$\sigma$	<code>\sigma{Space}</code>	<code>\S</code>	$\Sigma$	<code>\Sigma{Space}</code>
<code>\l</code>	$\lambda$	<code>\lambda{Space}</code>	<code>\L</code>	$\Lambda$	<code>\Lambda{Space}</code>
<code>\t</code>	$\theta$	<code>\theta{Space}</code>	<code>\T</code>	$\Theta$	<code>\Theta{Space}</code>
<code>\p</code>	$\phi$	<code>\phi{Space}</code>	<code>\P</code>	$\Phi$	<code>\Phi{Space}</code>
<code>\o</code>	$\omega$	<code>\omega{Space}</code>	<code>\O</code>	$\Omega$	<code>\Omega{Space}</code>
<code>\g</code>	$\gamma$	<code>\gamma{Space}</code>	<code>\G</code>	$\Gamma$	<code>\Gamma{Space}</code>

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

Matrix

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

Modifiers

Code	Output	Source
<code>\d1</code>	$\dot{x}$	<code>\dot{Space 2}</code>
<code>\d2</code>	$\ddot{x}$	<code>\ddot{Space 2}</code>
<code>\d3</code>	3 dots above	<code>\dddot{Space 2}</code>
<code>\d4</code>	4 dots above	<code>\ddddot{Space 2}</code>
<code>\~</code>	$\tilde{x}$	<code>\tilde{Space 2}</code>
<code>\v</code>	$\vec{x}$	<code>\vec{Space 2}</code>
<code>\h</code>	$\hat{x}$	<code>\hat{Space 2}</code>

Arrows

Code	Output	Source	Code	Output	Source
<code>\lr</code>	$\leftrightarrow$	<code>\leftrightharrow{Space}</code>	<code>\lrs</code>	$\Leftrightarrow$	<code>\leftrightharrows{Enter}{Left}</code>
<code>\la</code>	$\leftarrow$	<code>\leftarrow{Space}</code>	<code>\La</code>	$\Leftarrow$	<code>\Leftarrow{Space}</code>
<code>\down</code>	$\downarrow$	<code>\downarrow{Space}</code>	<code>\up</code>	$\uparrow$	<code>\uparrow{Space}</code>
<code>\ul</code>	$\nwarrow$	<code>\nwarrow{Space}</code>	<code>\ur</code>	$\nearrow$	<code>\nearrow{Space}</code>
<code>\dl</code>	$\swarrow$	<code>\swarrow{Space}</code>	<code>\dr</code>	$\searrow$	<code>\searrow{Space}</code>

Symbols

Code	Output	Source	Code	Output	Source
<code>\de</code>	$^\circ$	<code>\degree{Space}</code>	<code>\st</code>	$\star$	<code>\star{Space}</code>

## Structures

Code	Output	Source
<code>\r</code>	$\square$	<code>\right.{Left}</code>
<code>\ceil</code>	$\lceil$	<code>\lceil{Space}\rceil{Space 2}{Left}</code>
<code>\floor</code>	$\lfloor$	<code>\lfloor{Space}\rfloor{Space 2}{Left}</code>
<code>\brak</code>	$\langle$	<code>\bra{Space}\ket{Space 2}{Left}</code>
<code>\ls</code>	$\square P$	<code>^_ P {Left 4}</code>
<code>\fu</code>	myfunction $\square$	<code>\funcapply</code>

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

## Prefix for Fancy Letters

Code	Output	Source
<code>\sc</code>	$\mathscr{X}$	<code>\script</code>
<code>\do</code>	$\mathbb{X}$	<code>\double</code>
<code>\fr</code>	$\mathfrak{X}$	<code>\fraktur</code>

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

## 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](https://github.com/gtj1/symbol_assist)
- Intuitive Vim-like text cursor control: <https://github.com/RUSRUSHB/AutoTextCursor>

## Code Editing Guide

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

The code of `symbol_assist_OneNote.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`.

I added some parameters `co?`:

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