

Window 系统键盘流效率篇

工作会用 window 系统，建议学点 `AutoHotkey` 提升键盘操作体验。

AutoHotKey 相关资料：

- [官网](#)
- [下载](#)
- [中文文档](#)

另外推荐一款 window 下的神器 [listary](#)，用于体验 Mac 版的帽子。

有了解过的同学，可以直接参考下 [我的配置](#)。

能做什么

下面是我整合的一些常用操作：[github 地址](#) PS: 当然不一定是你想要的。而且很可能不习惯，所以在用之前你可以先大概看一下，最后修改为你自己想要的 ahk 脚本。

- 快速打开常用网址，如按下 f4 即可打开我的 github

Bash

```
1 f4::Run http://github.com/leer0911/
```

- 搜索引擎整合，如按下 f1 可以用谷歌搜索选中的文本，f6 可以触发搜索 前端常用的 [devdocs](#) 如 mac 里面的 `dash`

Bash

```
1 f1::
2 current_clipboard = %Clipboard%
3 Send ^c
4 ClipWait, 1
5 Run http://www.google.com/search?q=%Clipboard%
6 Clipboard = %current_clipboard%
7 return
```

- 光标快速移动，映射了键盘上的方向键。(把 alt 键当 mac 键用) 但是如果是用 vscode 编辑器。我还是觉得 `jumpy` 插件最好跳转了。用的不开心还能改源码~

- alt+i 触发 向上键
- alt+j 触发 向左键
- alt+k 触发 向下键
- alt+l 触发 向右键

Bash

```
1 ; 键盘键位替换,如 上下左右 删除 全选
2 &lt;!i::send {Up}
3 &lt;!k::send {Down}
4 &lt;!j::send {Left}
5 &lt;!l::send {Right}
6 &lt;!u::send {Backspace}
7 &lt;!a::send ^a
```

- 快速选择文本，用过 vim 的应该都喜欢光标快速跳转，文本快速选中。只要你想要。完全能打造成任意编辑器下的 vim。

- alt+shift+i 触发 向上选中文本
- alt+shift+j 触发 向左选中文本
- alt+shift+k 触发 向下选中文本
- alt+shift+l 触发 向右选中文本

Bash

```
1 +!i::send +{up}
2 +!j::send +{left}
3 +!l::send +{right}
4 +!k::send +{down}
5 +!;::send +{end}
6 +!h::send +{home}
```

- 一些常用但是按起来特别变扭的符号 如 # . 等，用别的按键映射。这个完全看个人习惯。如我不常用 CapsLock(大小写键) 则将其修改为 右 shift 键。

- CapsLock + j 对应 #
- CapsLock + d 对应 .
- CapsLock + k 对应 ()
- CapsLock + l 对应 { }

Bash

```
1 ; 常用符号快捷键生成 如 &lt; $ # * &gt; + ^ - = ! {} () . ,  
2 &gt;+h::send &lt;;  
3 &gt;+s::send $  
4 &gt;+j::send {#}  
5 &gt;+x::send *  
6 &gt;+u::send &gt;;  
7 &gt;+o::send {^}  
8 &gt;+i::send {+}  
9 &gt;+n::send -  
10 &gt;+m::send {=}  
11 &gt;+l::send {{}  
12 &gt;+k::send (  
13 &gt;+g::send {!}  
14 &gt;+d::send .  
15 &gt;+f::send {,  
16 &gt;+space::send {=}
```

- 我最常用的功能应该是 任意软件 置顶， 改变软件透明度。对于我们前端来说。总觉得屏幕不够用，老板又不肯买两个屏幕。怎么办。编辑器半透明呗，这样就可以边敲代码边看设计稿了。看 API 的时候也是。可以让编辑器置顶，半透明再配合 浏览器的 `vimium` 插件，你就可以边敲代码边看文档了。
 - 双击 shift 实现软件置顶
 - alt ++ 实现降低软件透明度
 - alt + - 实现增加软件透明度
 - 更多的功能，待你们去挖掘。记得分享~

觉得可以的话，可以继续看下教程。

快速入门

毕竟学习一门新语言是需要时间的，下面只是简单描述下我用 autohotkey 接触到的东西，这样当你想用我这套 hotkey 的时候能看懂并能修改成自己想要的。当然，想系统学习的同学可以直接参考[官方文档](#)

- 下载&安装 autohotkey
- 新建 后缀为.ahk 的文件并运行

热键

首先了解，代码中修饰符都代表了键盘中的哪些按键，你只要知道了修饰符边可以通过代码来让按键搞事情(脚本动作)。热键是通过一对 :: 创建的. 按键名或组合按键名必须在 :: 左边. 代码则跟在后面，以 Return 结束。(单行可省略 return) 比如

```
Bash

1 ^j::
2     Send, My First Script
3 Return
```

当你的 autohotkey 正常运行你编写的代码时。按下 `ctrl+j` 会输出 `My First Script` 这么一段文字。

符号	说明
#	win
!	alt
^	control
+	shift
&	用于组合按键
<	左按键
>	右按键
*	通配符
~	保留按键原功能
UP	释放时触发热键

[详细](#)

禁用按键

通过不为按键或按键组合的热键指定任何操作可以在整个系统中完全禁用它们. 下面的例子中禁用了右 Windows 键:

```
Bash

1 RWin::return
```

上文相关按键

当需要在某些特定软件中设定热键则需要了解，上下文按键

[#IfWinActive/Exist](#) 和 [#if](#)指令可以用来让热键根据不同的条件执行不同的动作例如:

Bash

```
1 #IfWinActive, ahk_class Notepad
2 ^a::MsgBox 你在记事本中按下了 Ctrl-A . 而在其他窗口中按下 Ctrl-A 将原样发送.
3 #c::MsgBox 你在记事本中按下了 Win-C 组合键.
4
5 #IfWinActive
6 #c::MsgBox 你在非记事本程序中按下了 Win-C .
7
8 #If MouseIsOver("ahk_class Shell_TrayWnd")
9 WheelUp::Send {Volume_Up} ; 在任务栏上滚动滚轮:增加/减小音量.
10 WheelDown::Send {Volume_Down} ;
```

热字符串

热字符串主要用于扩展你的缩写库(自动替换). 当然, 它也可以用来映射任何脚本动作。比如

Bash

```
1 ::ftw::Free the whales
```

热字符串会将你输入的"ftw"转换为"Free the whales"。

搞事情啦

知道了热键和热字符串，肯定想知道能用他来搞什么事情。

发送按键

现在你决定发送一些按键到一个程序中. 你可以使用 Send 命令. Send 表示发送按键, 模拟打字或按键操作.如

Bash

```
1 Send !+a
```

会按下 ALT+SHIFT+a

[更多 send 命令](#)

运行程序和网页

想要运行画图(mspaint.exe), 计算器(calc.exe), 脚本 .ahk 等程序或要打开一个文件夹, 你可以使用 Run 命令. 你还可以用这个命令打开一个网址, 比如打开 <https://autohotkey.com/> . 如果你想打开一个已经安装好的程序, 也很简单, 就像这样:

Bash

```
1 ; 运行一个程序. 注: 大部分的程序可能需要完整路径.
2 Run, %A_ProgramFiles%\Some_Program\Program.exe
3
4 ; 打开一个网址
5 Run, https://autohotkey.com
```

[更多 run 命令](#)

命令和函数索引

AutoHotkey 有两个重要的工具供开发者使用:命令和函数

[官方文档](#)