

23 | 效率工具：选对用对才能事半功倍

2019-10-16 葛俊

研发效率破局之道

[进入课程 >](#)



讲述：葛俊

时长 15:05 大小 13.82M



你好，我是葛俊。今天，我来和你聊一聊效率工具。

在前两篇文章中，我分别从编程技术和工作方式两个方面，与你介绍了如何提高个人研发效能，成长为 10x 程序员。今天，我再与你分享些工具使用方面的技巧和方法，来帮助你全方位地提高自己的工作效率。

谈到研发效能，一定离不开工具。工具使用得当，可以大幅提升研发效率，甚至可以因此引发业务发展、开发行为的质变。比如，我在[第 11 篇文章](#)中与你分享的截屏工具链带来的效率提升，就使得其被广泛用在了 Commit Message 中，大幅提高了整个过程的效率。

我一直对各种工具很感兴趣，从业这些年来，在这上面花了不少时间，也踩了一些坑。比如，我曾经在 Emacs 上面花费了大量的时间，去自动化和优化很多细枝末节的操作。结果

是，花在优化和自动化上的时间，比节省下的时间还要多，得不偿失。

所以，我总结得出，**对于工具一定要选对和用对。**

选对工具指的是，我们要针对不同的任务，找到合适的工具来提高效率。而用对工具指的是，我们要分配适量的时间和精力来选择工具，要时刻注意投入产出比。

关于用对工具，我推荐一个比较好的方式是，**留意工作中经常重复和繁琐的工作步骤，隔一段时间就做些工具方面的调研，看看有没有更合适的工具或者使用方法来优化这些流程。**对于一些非常繁琐、执行频率又非常高的操作，如果没有现成的工具，甚至可以考虑自己开发一些工具和脚本进行优化。频次越高的操作，越值得优化。

接下来，我就按照功能和你分享一些有助于提高工作效率的工具。

因为我最近的工作电脑是 macOS 系统，所以我今天推荐的适用于苹果操作系统和生态的工具会偏多一些。但，这些优化的方向和思路是一致的，在 Windows 操作系统上你也能找到类似的工具。另外，安卓系统更加灵活，有比苹果生态更多的工具。

第一个任务类别是，操作系统上的通用操作

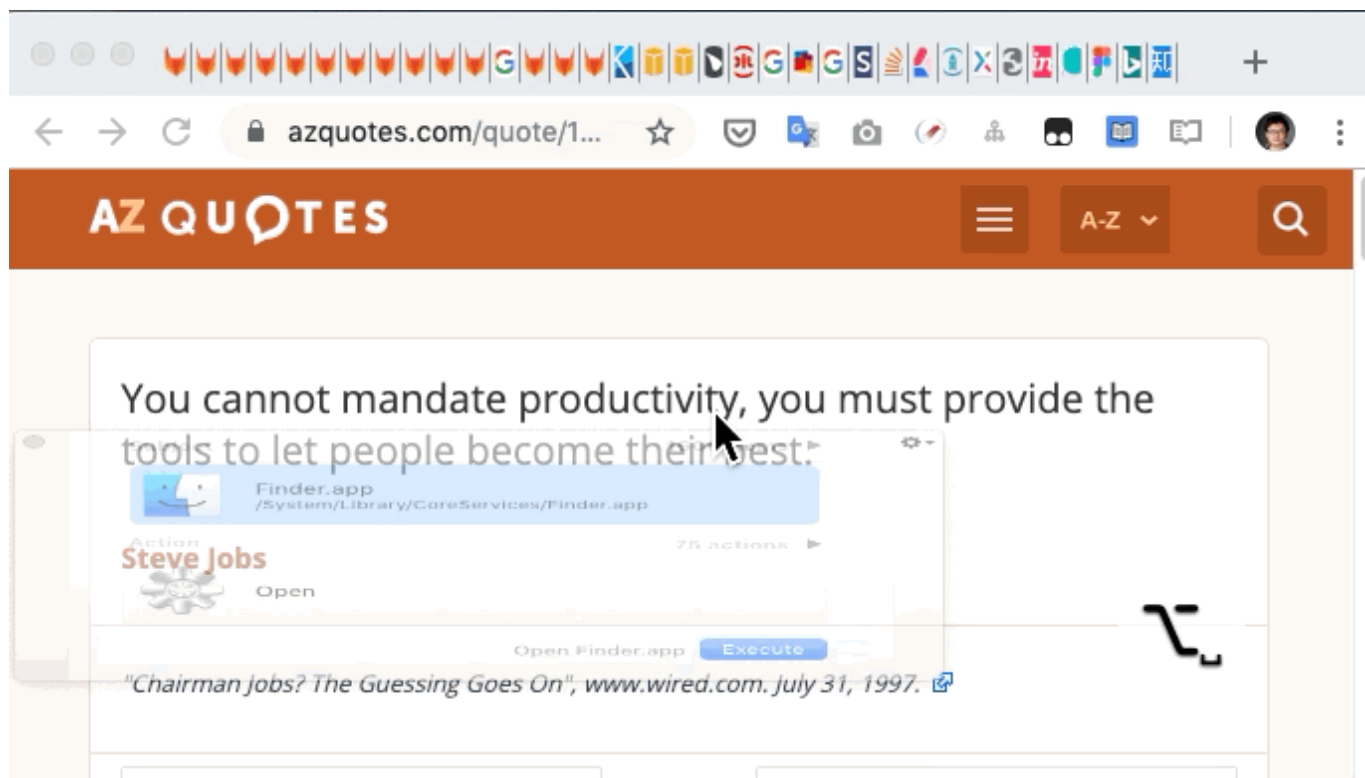
操作系统上的通用操作，主要有窗口切换、程序启动、窗口管理、剪贴板管理等。这些操作普遍且频繁，比较值得优化。

关于窗口切换，各操作系统都有一些自带的支持。比如，macOS 系统自带的方式是 cmd+tab，Windows 系统自带的是 Alt+Tab 或者 Win+Tab。**关于程序启动**，macOS 系统有 Spotlight，Windows 系统可以使用 Win 键或者 Win+R 启动任务。

但这两个操作实在是频繁，且操作系统自带的功能只是基础功能。所以。我推荐你进行进一步优化。

在 macOS 系统上，Alfred、Quicksilver 这两个工具就很不错。Windows 系统的话，我推荐 [Wox](#)。

这些工具的用法也很简单，使用一个快捷键启动，然后输入过滤条件，查找需要运行或者切换的程序，回车即可。使用 Quicksilver 启动 / 切换程序的方法，如下所示。

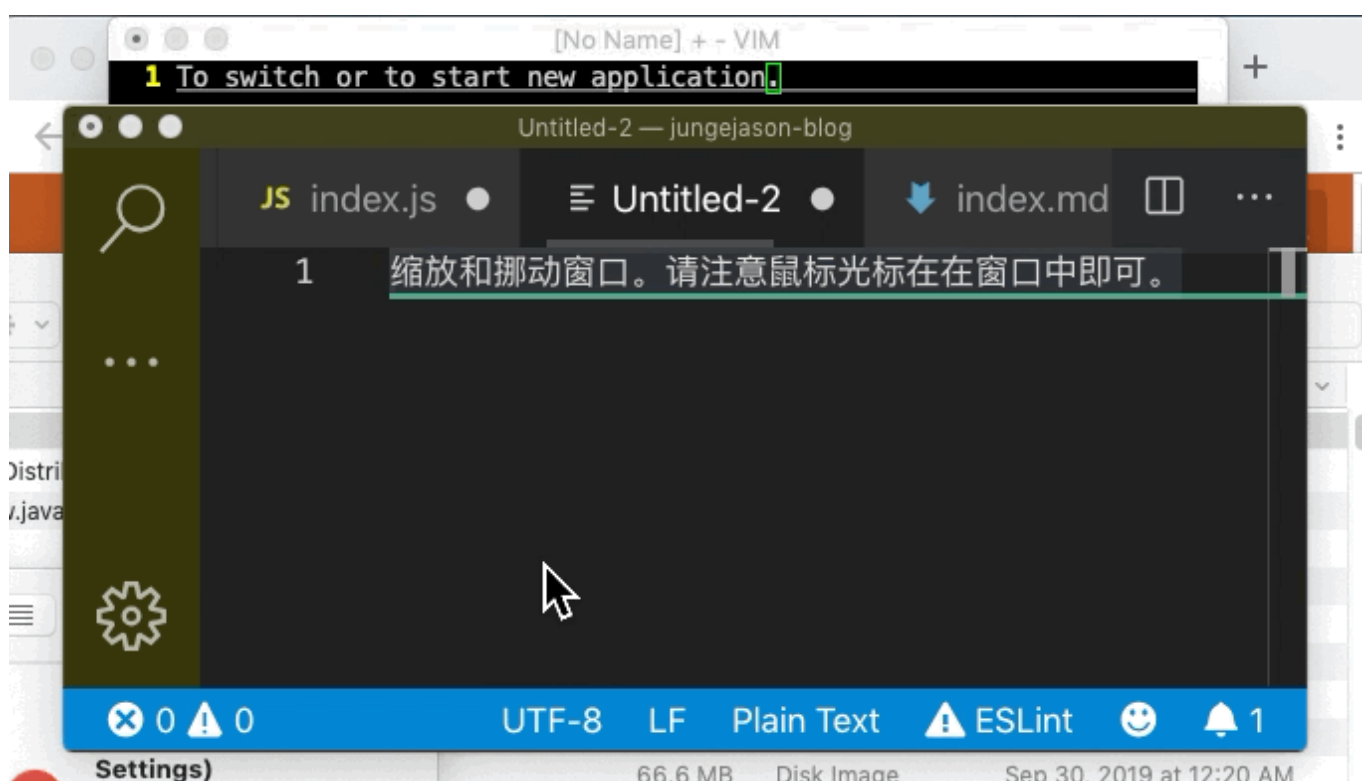
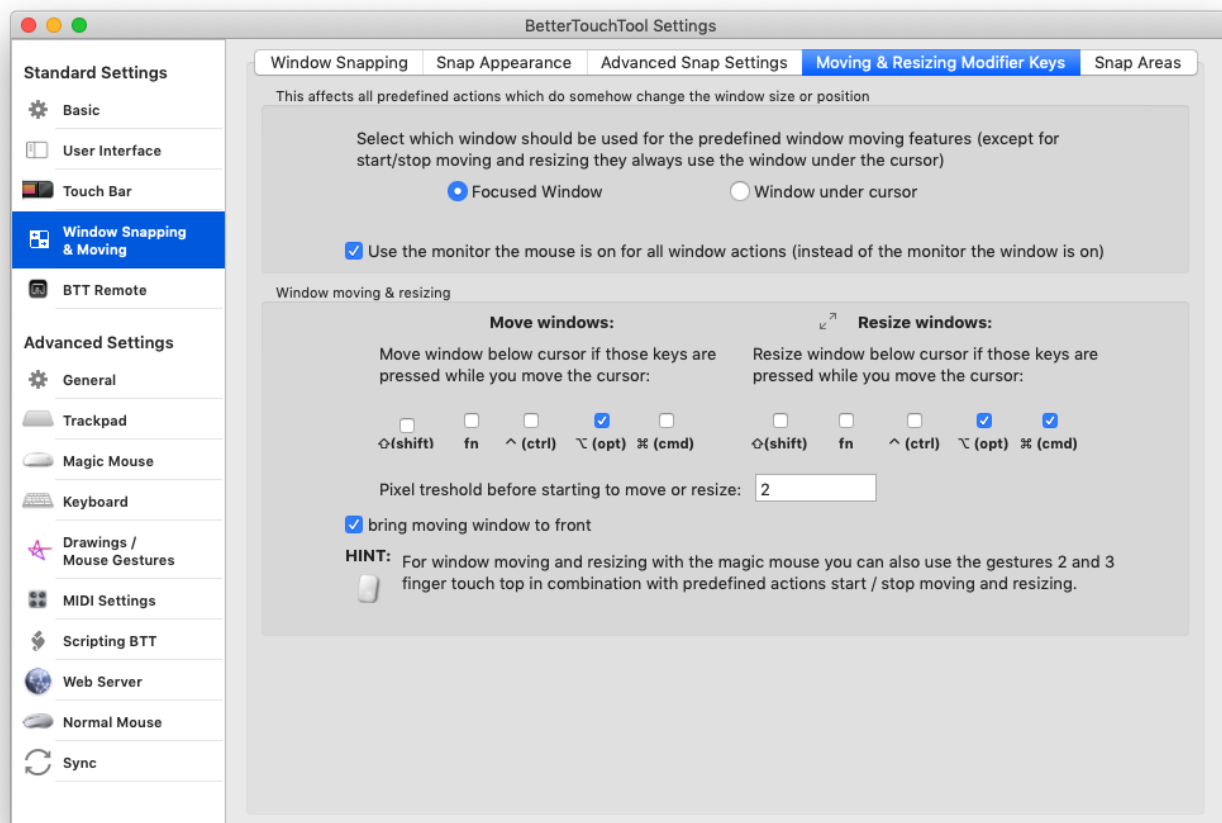


关于窗口管理，有一个常用操作，是把一个窗口自动缩放，并摆放到屏幕的某一个位置（比如屏幕的左上角）。

在 macOS 系统中，我一般使用[BetterTouchTool](#)这个工具实现，也推荐你使用。通过它，我可以非常方便地实现窗口缩放和拖动。有两个操作，我觉得特别方便：

- 一是，按住 alt 键，无论光标在窗口的任何位置，都可以通过挪动鼠标拖动窗口；
- 二是，按住 cmd+alt 键，无论光标在窗口的任何位置，都可以通过移动鼠标来改变窗口大小。

BetterTouchTool 的配置和使用，如下面两幅图所示。



最后是，**剪贴板管理**。目前，操作系统自带的剪贴板只能保存一条记录，但如果我们使用剪贴板历史管理工具的话，可以非常方便地保存和使用多条历史拷贝内容。

在 macOS 系统上，我用的工具是 Quicksilver。Windows 上类似的工具有很多，比如 [Ditto](#)。

第二个任务类别是输入


关于输入，我主要和你推荐 4 类工具，分别是语音输入、文字快速输入、重新定义按键和具体的键盘。

我们先看看第一类工具，语音输入。我在电脑上常用的是[讯飞语音输入](#)，手机上常用的是[讯飞输入法](#)。在电脑上使用语音输入有一个技巧是，把手机作为麦克风。这样输入的效果非常好，尤其是旁边有人说话的时候，手机麦克风的过滤功能就非常棒了。

在手机上，我经常使用讯飞输入法的语音便签功能，保存录音的同时还可以直接把语音转为文字，对于记录灵感来说特别方便。

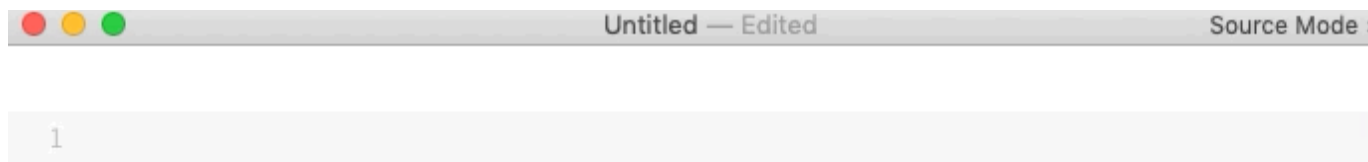
第 2 类工具是文字快速输入。Mac 上最流行的是[TextExpander](#)，它可以方便地生成预先定义的字符串、当天日期，也可以很方便地指定文字生成之后的光标所在位置等。

比如，经过下面的配置之后，只要输入缩写 “;mj” 就可以生成日记模板，并把光标放到第一个任务处。


 复制代码

```
1 ## %Y-%m-%d:
2 今天关键任务
3 %|
4 -
5
6 成功之处，不足之处
7 -
```





在 Windows 系统上，开源免费的[AutoHotkey](#)非常强大。它不但可以进行文本扩展，还可以运行程序。我来和你举一个运行程序的例子吧。比如，你可以使用 Win+V 直接切换到 VIM。如果 VIM 先前没启动的话，则会直接其启动。

 复制代码


```
1 ;;; switch to vim
2 #v::
3 if WinExist( "ahk_class Vim" )
4 {
5     WinActivate
6 }
7 Return
```

第三类工具是，重新定义按键。在我看来，对程序员最实用的一个功能是，把大写锁定键（caps lock）转换成 ctrl 键。因为 caps lock 的位置很方便使用频率却很低，而编程中常会用到 ctrl 键，所以这个重新定义对程序员非常有用。

至于重新定义按键的工具，在 Mac 上我使用的是[Karabiner-Elements](#)，在 Windows 上使用 AutoHotkey 就可以。

另外，我对 caps lock 键的定义不是简单地重新定义为 ctrl，而是如果和其他键共同使用的话就是 ctrl 键，单独使用的话就是 esc 键。因为，esc 键的使用频率也很高，尤其是在 VIM 中。

在 Karabiner-Elements 中的具体设置方法是：

 复制代码

```
1 "rules": [  
2   {  
3     "manipulators": [  
4       {  
5         "description": "Change ctrl to esc if pressed alone.",  
6         "from": {  
7           "key_code": "caps_lock",  
8           "modifiers": {  
9             "optional": [  
10              "any"  
11            ]  
12          }  
13        },  
14        "to": [  
15          {  
16            "key_code": "left_control"  
17          }  
18        ],  
19        "to_if_alone": [  
20          {  
21            "key_code": "escape"  
22          }  
23        ],  
24        "type": "basic"  
25      }  
26    ]  
27  }  
28 ]
```

第四类工具是，具体的键盘。开发者要经常输入，选择一款顺手的键盘就很重要了。键盘的选用，我看重的是手感和键盘布局。手感的话，机械键盘的确好一些，你可以根据自己的体验选择一款合适的。而布局的话，左右手按键分离较远的一般来说会好一些。

根据这些原则，我平时使用[Kinesis Pro2 键盘](#)。因为，它的中间是凹进去的，所以手指与每一行按键的距离都差不多，输入时很轻松。另外，特殊键由大拇指控制，这个特点我也非

常喜欢。



第三个任务类别是，知识管理

对学习和知识管理，我主要与你分享 6 类工具。

第一类工具是**云盘**，用来确保存储的内容不会因为本地电脑的意外情况而丢失。目前，市面上的各种云盘都可以满足这个需求。这里我有一个小建议，使用云盘时最好能够使用本地的自动同步工具，自动地将本地文件夹里的内容同步到云盘。

第 2 类工具是**笔记**。我挑选笔记工具的原则，包括支持云同步，支持电脑和手机端同时访问。印象笔记、Mac 自带的 Notes、石墨文档等都不错。

第 3 类工具是**写文档的工具**。相较于笔记工具来说，写文档的工具还有一个挑选原则是支持 Markdown。我现在是组合使用 iCloud、1Writer 和 Typora：使用 iCloud 进行同步，在手机使用 1Writer，在电脑上使用 Typora。

第 4 类工具是**思维导图**。对思维导图，比较重要的特性包括跨平台、方便使用快捷键、方便导入\导出其他格式。我以前一直用的是[FreeMind](#)，它基于 Java 实现，在 Windows、macOS 和 Linux 系统上都可以使用，快捷键配置也很强大，但它的一大缺陷是在手机上不能使用。所以，最近两年我转移到了[XMind Zen](#)上面。它可以在电脑和手机上同时使用并自动同步，显示也比较美观。

第 5 类工具是**截屏、录屏**。我对截屏、录屏工具的选用原则是标注功能要好，另外要可以把录屏保存成 GIF 格式。我现在最常用的是对这两方面都支持不错的[MonoSnap](#)。另外，录屏的时候，我们常常希望把当前的键盘输入显示到屏幕上，我推荐使用[KeyCastr](#)；编辑 GIF 文件的话，我使用的是[GIF Brewery 3](#)。

这个专栏中，你见到的录屏和截屏图片，我就是使用这三个工具完成的。另外，还有一个对命令行终端录制比较方便的工具[asciinema](#)。它可以把你终端里面的操作和输出用 JSON 形式保存下来，也可以上传到 [asciinema](#) 的网站上，还可以方便地在你自己的网站上引用显示。但这个工具的缺点是，无法显示键盘的按键操作。

第 6 类是**处理 PDF 文件的工具**。这样的工具我主要关注基本的标注功能，以及在平板电脑上可以使用电子笔操作。在电脑上，我一般使用[Foxit Reader](#)和 macOS 自带的 Preview；在手机上，我主要使用的是[PDF Expert](#)。

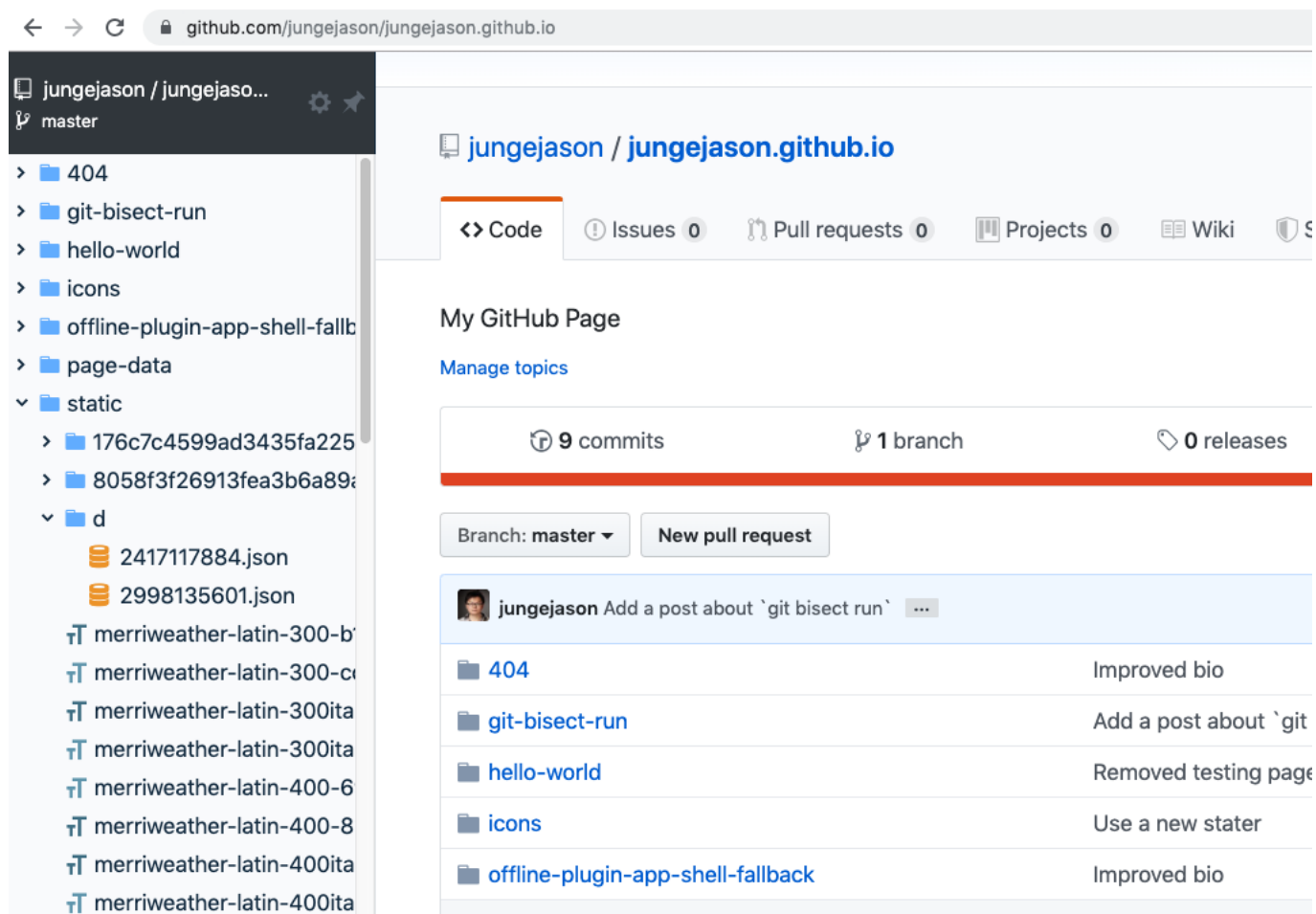
在我看来，一个比较高效的方式是，把 PDF 文件放到笔记软件中，然后在笔记中直接对其进行操作。印象笔记和苹果自带的 Notes 都支持这个功能。

第四个任务类别是，浏览网页

访问互联网是非常高频的操作，所以我把它单独列成一类。对开发人员，我推荐使用 Chrome 和 Firefox，因为两者都有很强的扩展功能，对开发活动比较友好。

因为最近几年我主要在使用 Chrome，所以接下来，我与你介绍一些我常用的 Chrome 插件和技巧。

第一个插件是[Octotree](#)。它会在你访问 GitHub 查看项目时，在窗口左侧以树形结构清晰地显示代码仓结构。



第二个插件是[Pocket for Chrome](#)。它需要与[Pocket](#)配合使用，实际上就是一个在线的网页书签保存服务，但它的推荐功能很优秀，并且对手机端的支持也很好。

第三个插件是[smartUp Gestures](#)。通过它，你可以按住鼠标右键不放，在屏幕上画一些形状，也就是手势（Gesture），来进行一些操作。

我最常使用的功能有：切换到左边（或右边）一个标签页、向下（或向上）翻页至网页末尾（或网页开头）、返回网页访问历史的上一页（或下一页），以及关闭当前标签页。

在文稿里我有一个录屏供你参考。

Jason's Blog



Written by Jason Ge 葛俊

Git Bisect Run

October 01, 2019

Describe how git bisect run works



第 5 个任务类别是，开发

和开发工作相关的工具比较多，包括编辑器（比如 VS Code、VIM）、IDE（比如 Visual Studio 和 JetBrains 系列）、代码仓管理工具（比如 Git 和 HG）、API 测试工具（比如 Postman），以及命令行工具等。

关于开发工具的使用，我会在后面的文章中与你详细介绍其中的使用技巧，帮你提高编码效率。

小结

无论“磨刀不误砍柴工”“工欲善其事，必先利其器”，还是乔布斯曾说过的“You cannot mandate productivity, you must provide the tools to let people become their best.（你不能强制要求大家提高生产力，你必须提供工具，让大家发挥他们的最大能力）”，强调的都是工具的重要性。

所以，作为开发人员，我们一定要选对工具、用对工具，才能提高效能。在今天这篇文章中，我从操作系统上的通用操作、输入、知识管理、网页浏览和开发 5 个方面给你推荐了一些工具及使用方法。当然了，关于开发工具这个重头戏我还会在后面的文章中与你详细分享。

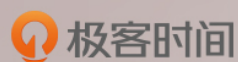
其实，使用哪款工具，是一个仁者见仁，智者见智的问题。所以，通过这篇文章，我更希望达到的目的是，能帮助你对日常工作中最常见的操作进行思考，寻找值得优化和可以使用工具优化的地方，从而提高个人的研发效能。

但，我还要强调的是，工具只是辅助，编程工作更重要的还是思考。所以，我建议你不要花费过多的时间在工具研究上，要时刻留意投入产出比。

思考题

除了今天提到的操作系统上的通用操作、输入、知识管理、网页浏览和开发 5 个方面外，你觉得还有哪些方面的操作比较频繁吗，有什么值得优化的地方吗，又有什么值得推荐的工具吗？

感谢你的收听，欢迎你在评论区给我留言分享你的观点，也欢迎你把这篇文章分享给更多的朋友一起阅读。我们下期再见！



研发效率破局之道

Facebook 研发效率工作法

葛俊

前 Facebook 内部工具团队 Tech Lead



新版升级：点击「👤请朋友读」，20位好友免费读，邀请订阅更有**现金**奖励。

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 22 | 深度工作：聚焦最有价值的事儿

下一篇 24 | VIM：如何高性价比地学习VIM的实用技巧？

精选留言 (7)

写留言



Jxin

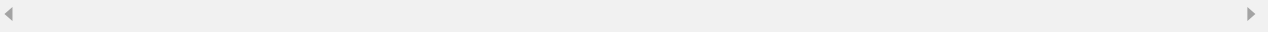
2019-10-16

win的一些工具:

- 1.Listary 全文检索
- 2.Beyond Compare 文件比较
- 3.cmder 支持linux命令的命令框
- 4.SwitchHosts hosts工具...

展开 ∨

作者回复: 多谢分享!!



2



Y024CN

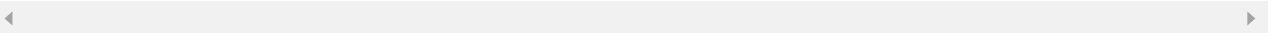
2019-10-16

为每天遨游于各种 API 文档、代码片段的 yuan 们, 安利一款工具:

Mac 下 Dash: <https://kapeli.com/dash>

Windows 下 Zeal: <https://zealdocs.org/>

作者回复: 我有装过, 后来发现还是用网页搜索更直接。当然这个纯属个人偏好:)



1



花生无翼

2019-10-16

工欲善其事必先利其器

展开 ∨



1



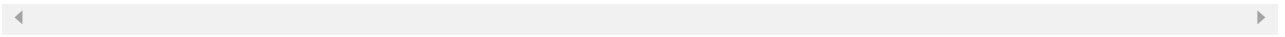
张裕

2019-10-18

tmux, fzf, bat, ripgrep 配合neovim很好用

展开 ∨

作者回复: 我还没有用过neovim。改天试试!



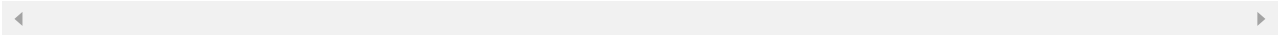
JohnT3e

2019-10-18

浏览器可以使用vim模式插件：Chrome(Vimium)或者Firefox(Vimium-FF)

作者回复: 我在使用Firefox的时候使用过类似的VIM插件。不过发现还是需要用到鼠标，最后就干脆没有这样的插件了。

我有朋友一直在用，他很喜欢。



许童童

2019-10-17

程序员一定要学会使用科学上网

展开 ∨



李双

2019-10-16

磨刀不误砍柴工👉

展开 ∨

