

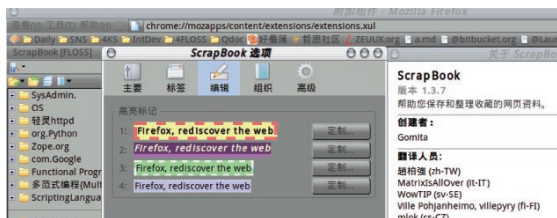
# 我的工具箱

■ 文 / Zoom.Quiet

**现**在想来，对于工具好象是追求：MOO（Muscle Oriented Operating，面向肌肉的操作），即仅仅键盘操作就可以达到目标。毕竟节省从鼠标定位或菜单操作中的零点几秒时间，在日常大量操作积累后，就是一笔可观的收获了，而且形成肌肉记忆后，将会不断地加强，不会因为环境的变化而失效，这样的工具才是终身受用的。下面根据自己生产活动中的主要环节，来说明选用的工具组合。

## 收集

资料的收集是一切的起点，如今信息形式电子化，其中最多的就是网页，所以对于俺最重要的就是网页的收集管理。最初是使用分类目录收纳，但问题在于无法快速定位。主力浏览器迁移到火狐后，也体验过资料管理器Zotero，都难以达到功能、性能、管理三者均衡，最终还是坚持了ScrapBook，它是Firefox的插件工具，是最简单轻便的网页管理工具，仅用右键点选菜单即可抓取指定内容到本地，并组织到指定的分类中，同



网页管理工具ScrapBook

时提供足够可用的功能，方便进行各种二次处理。还可以自动输出网站样HTML索引页面，俺个人网站的资料分享（<http://floss.zoomquiet.org>）就是直接使用这一功能发布的，需要注意的是，一定要有比较稳定和精简的知识体系分类，自个儿对任何一技术领域的资料，可以习惯性的完成分类和安置，这才在日后能够随时快速查询到。

## 思考

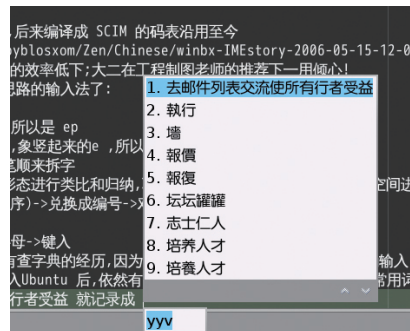
收集后是面对大量资料的思考阶段。Freemind是常见的跨平台思维图谱工具，支持格式规范的XML，可以用XLS等工具进行二次处理。但我无法忍受它在复杂图谱中超肉的反应和不时意外崩溃。现在逐渐统一使用Graphviz，通过编辑.dot图形脚本来记录思维，用Graphviz提供的各种工具（比如说命令:dot）输出图形来展示思维，达到了禅一样的高度。而且可以输出图片热区定义MAP文本，自然部署到HTML中成为可点击图片。更加厉害的是Graphviz可以集成到各种工具中融入日常工作。例如：文档化开

发注释提取工具Doxygen中就内置了Graphviz来对代码的类关系进行图形化展示。以往的图谱工具都是GUI界面的，都需要鼠标操作，强迫我们在叙述思维关系时，要关注排版

的美观，只有使用Graphviz工具时，通过.dot脚本用纯文本，通过字符的简单约定来表述的思维关系，可以自然地使用版本系统进行统一管理，追踪变化。同时，可以随时调用命令生成图片，完美的解决了思维图谱的维护和展示两个完全不同层面的行为。Graphviz并不强行要求使用.dot脚本，也不需要任何特定编辑工具，只要可以编辑文本的软件都可以来“画”思维图谱。

## 输入

输入法可以说是最频率使用的软件了，俺使用BXM（表形码），来自Windows 95SE内置输入法，一用倾心，最后编译成SCIM的码表沿用至今。它的精巧在于严格依照自然的汉字间架以及笔顺来拆字，同时将英文字母的形状和中文部首的形态进行类比和归纳，直接对应成码表。不象其它形码类输入法，只是将键盘空间



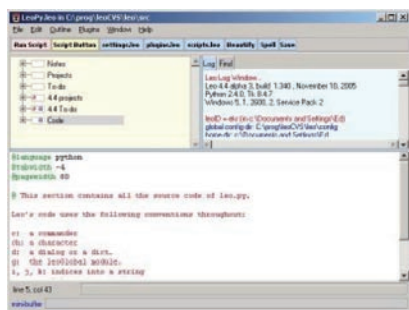
用表形码将快速调用常用词语

进行编码。承SCIM的情，BXM导入Ubuntu后，依然有完备的自定义功能，随时可以将近期常用词语记录成字典，快速调用。

日常写文档，少不了截屏工具，但并不是功能越多越好，著名的Snapit就一直没有用起来。在Windows中，最习惯的截屏工具是SPX，喜欢其快捷键操作，图片有撕边儿效果，功能极少，但都刚好需要。

迁移到Ubuntu后就用命令行工具了：scrot -s -d 1 -e 'mv \$f ~/4snap/zoomq-\$f' 将这行命令设定成桌面快捷方式，就完成了习惯的截屏操作迁移，可以用管道将图片输出给其它图片效果处理脚本，获得一样的撕边儿效果。

当Coder当然要有趁手的编程环境，俺主力编辑环境是Leo（辅助编



文学化编程的得力助手Leo

辑器是最土的Gedit）。曾逐一尝试过各种编辑器，都无法沉浸其中。最终遇到并理解了Leo后，就无法放手了。很难描述为什么，Leo本身是种很难定位的编辑器（因为其独特的文学化编程理念，完全颠覆了我们习惯的所有IDE行为）。理论根源于Donald Knuth针对结构化编程，在撰写TAOCP时发明的：文学化编程（Literate Programming），就俺的体验：就是将编程与写作视作同种行为，认为文本的层次和结构要比语法结构重要！是真正解放代码和思想间隔阂的生产力工具。最核心的内容爽直地说是：“可以直接

将思维层次和文本对应起来！”

一般我们编程时，在IDE帮助下，整体上面对的还是一维的面向单独文件的平铺下来的文本流，而Leo通过在界面中树形节点的操作配合文本编辑，将程序的自然思维形式化了，同时又通过丰富的输出方式，确保了这种基于节点树的形式化和编译器要求的线性文本流兼容！优雅的实现文学化编程。

各种文档的撰写算是代码之外工作生活的基本输入形式。在WYSIWYG（所见即所得）的界面中，我们被迫不断地在作者、排版、美术、编辑角色间跃迁，其实被浪费了太多精力，后来才知道世界上有WYTIWYG（所想即所得）的文档组织过程，一旦理解，就陷入无法割舍的享受。一切都是结构化文本的魔力。即通过简单的空格或是字符约定，标识出文章结构，撰写时，完全不考虑排版啊字体啊什么乱七八糟的事儿，专注将心中所想，明确的内容结构表述出来，具体的展示由专门的工具自动处理，输出成标准化格式文档。

这方面，我习惯使用reStructured Text（缩写为rST），不论是文章还是作为大型文档的组织核心格式，都可以担当，其中Sphinx提供了依托rST的完备图书工程组织工具平台。这是支持模板的一整套图书工具，可以将一组rST根据配置，生成HTML/CHM/PDF。有完备的命令行工具支持，结合版本管理系统，可以轻松的在任何服务器上架构自动化图书/大型文档编译事务。

## 其他日常工具

大家可能发现了，俺是自由软件狂信者，从06年开始使用Ubuntu，所有工具全部是优先面向GNU/Linux的，但以上工具除了Scrot可都是跨平

台的。

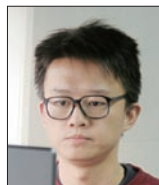
在紧张的工作中，长期以同一姿势定在电脑前，非常容易引发RSI（Repetitive Strain Injury，重复性压力伤害），Workrave这个软件是协助预防与复原RSI！可以使用Workrave强行提醒俺定期离开电脑前，放松身体，并提供保健操指引，有助于形成固定的高效工作节奏。

工具箱依然在补全中，但不论什么电脑工具都无法替代纸笔，都不如其高效。但是，这一组合的使用又实在太考验使用者，俺现在的初步使用方式是：A6幅面的80页线圈本，配普通水笔。从前到后，是每日时间帐单：两面一页一天，左边是顺序的任务用时记录，右边是当日计划任务图谱分析。从后到前，是沟通纪要，坚持使用思维图谱形式。主要问题是：定期的时间支出统计和任务完成经验整理，有个从纸到电脑的转换。这一过程还没有找到靠谱的工具来自动进行。

## 结束语

对于俺，工具的选择与组合是将近14年的连续过程，很艰难地才算形成一点当前的原则，只是强调所有工具都不是孤立存在于个人工具箱的，而是作为整体服务于俺的生活与工具领域。让我们面向文本记录，面向数据程序，面向肌肉操作吧！P

## 作者简介



Zoom.Quiet，Pythoner。中文Python用户组创始人，哲思自由软件社区核心成员，Erlang 中国用户组宣传部长，同为教育大发现社区高级顾问。

■ 责任编辑：常政 changzheng@csdn.net