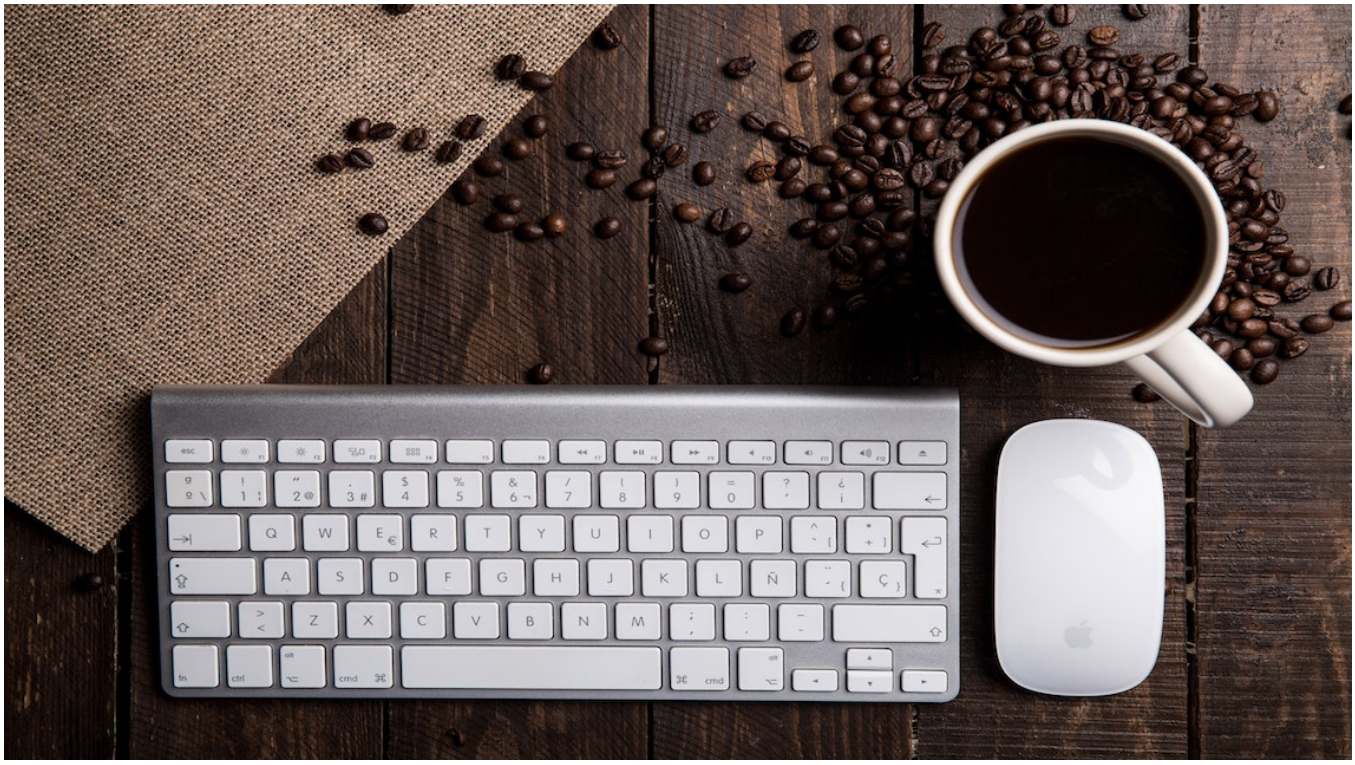
01讲学编辑器，到底应该学什么



你好，我是吕鹏，今天这篇文章是我们整个专栏的第一课，所以在这里我先向你问个好。我想在接下来的这三个月里，你我肯定是亦师亦友，共同切磋，一起进步。

极客时间团队经常提到一个词叫“学习路径”，我觉得特别有意思。同样，对于编辑器而言，我在开学第一课里，不应该深入到具体细节里，而是应该告诉你，怎么才能快速熟练掌握一个新的编辑器。

在我看来，编辑器的学习，和编程语言有一点类似。你当然可以找一本官方手册，从头到尾事无巨细全部学习一遍，这里面包括基本语法、数据类型、控制结构、函数、设计模式、框架等等。一开始在你不熟悉这门编程语言的时候，你可能会陷入到某一个语法的细节里很久不能自拔，但最后等你掌握了之后，你才发现，当初自学时自己纠结的点根本不影响大局，那些你当初认为重要的细节其实一点不那么重要。

在掌握了第一门编程语言之后，你想要学习另外一门新的编程语言，如果这个时候，你的学习路径还像学习第一门编程语言时那样，那我可以武断地说，你并没有从全局上理解这门编程语言，也没有在脑海中建立起“学习框架”。

怎么说呢？其实编程语言在设计上大同小异，新的编程语言往往都是在解决老的语言的某一个短板，但他们在最本质的设计上不会有大的变化。在有了这个框架之后，你再去学习，就不至于“拣了芝麻丢了西瓜”了。

同样，编辑器的学习，也和编程语言一模一样。

当使用一个工具时，你关心的应该是它能用来做什么，它最擅长做什么，以及它做不了什么，换句话说，就是这个工具的“下限”和“上限”。

首先，我们一起来找下编辑器或者 IDE 的“下限”。一款开发工具的“下限”是由它默认自带的功能决定的，也就是它开箱即用时的体验。对于大部分用户而言，工具的“下限”也就决定了他们会不会马上卸载它。

1. **快捷键的选用**

说到编辑器的自带功能，你最先想到的肯定是代码编辑的快捷键，快捷键的设计直接决定着这个编辑器用起来是不是顺手，效率到底高不高。

我们先从编辑器开发者的角度来看这个点。一般来说，编辑器的开发者要尽可能保证这些快捷键的默认配置接近大家的使用习惯。比如 VS Code 在给新功能选择快捷键时，会先看看系统是不是有类似的功能，以及系统用的是哪个快捷键组合。

举个简单的例子，在 macOS 上，按住“Cmd +delete”就能够把当前行光标前的所有内容都删掉，如果我们在 VS Code 里给 “Cmd + delete”配上别的功能，用户可能就要生气地扔键盘了。

除了看系统上的惯用快捷键，开发工具之间也会互相参考。最后，也是最重要的一点，我们要看快捷键的配置是否有统一性。Shift 键能用于控制文本选择的，Ctrl 或者 Cmd 键能当做辅助键的，Tab 是用于在控件之间跳转的……也就是说，如果编辑器在选择快捷键的时候使用了某些规则，那么就要坚定地贯彻下去，否则就会给用户带来困扰。

理解了编辑器开发者实现快捷键特性的思路和出发点之后，你再来学习使用编程工具就有章法了。这里我再插一句，学习很多工具的时候，你都可以想想，如果你是设计者，你会怎么做，顺着这条主线去思考问题，你应该会更容易理解为什么是这个“现状”。

所以，对于快捷键的学习，你可以先看看默认的快捷键有哪些是标准的、主流的，快捷键之间有没有什么内在的规则。然后，你还要试着找出快捷键的分类方式。毕竟功能总是比键帽的数量多，所以能够把这么多功能映射成快捷键本身就是门学问，了解了这门学问后，你再来掌握所用编辑器的快捷键，就会轻松很多。

VS Code 的做法是搭配组合辅助键 Ctrl、Alt 和 Meta 键，然后根据当前用户正在使用的组件来决定是什么功能；Vim 则是运用了“模态”的概念，把输入文本信息，和光标移动、文本操作等分开为不同的模态，不同模态各有一套自己的快捷键。大家都说 Vim 的学习曲线比较陡峭，一个很重要的原因就是“模态”的概念实在是另辟蹊径。

就我和很多编辑器爱好者的交流来看，掌握一套设计良好的快捷键，是折腾编辑器的过程中最划算的投资。大部分编辑器都支持自定义快捷键，这样你就可以一直使用自己熟悉的那一套了。

Vim 都快30岁了，现在哪个开发工具要是说没法支持 Vim 的操作方式，那真的会相形见绌。投资一种高效的快捷键操作，花点时间让它成为你的肌肉记忆，然后用30年，这样你的整个职业生涯都会受益。

2. **编程语言与框架支持**

除了快捷键，另外一个决定着编辑器“下限”的就是编程语言和框架的支持了。前面我提到过，我们需要根据编程语言来选择相应的编辑器。

如果你得写 iOS 应用，绝大多数情况要用 Objective-C 或者 Swift，那你很难脱离 Xcode；如果公司是用 ASP.NET，你需要写 C#，Visual Studio 就是现成最好的IDE；但如果你们用 Ruby on Rails 做网站，而 Ruby 社区又没有钦定的官方开发工具，Vim、TextMate、RubyMine 这些都有人使用，各有各的特点，你从中选择一款就可以。

因此，当你去考察一个新的编辑器时，就需要看它对你必须使用的编程语言的支持情况如何，语法高亮是否正确，是否有自动补全功能，能否直接调试和运行测试。你甚至需要研究怎么让它支持你电脑上特殊的环境变量，怎么更好地和你们使用的框架集成，这些就是你要付出的学习成本。

3. **对工作流的选择和支持**

最后一个影响因素，就是编辑器对工作流的选择和支持。工作流的选择，既是态度，又非常能够体现时代背景和行业现状。

好比说 Vim 并不自带资源管理器，如果你希望在 Vim 里看到文件树并快速浏览切换文件，就需要安装额外的插件。现在软件开发工程化比较成熟了，开发工具也都直接集成了资源管理器；VS Code 自带了版本管理，可以说是对版本管理在软件开发过程中的重要地位的一种认同；但有些开发工具把测试功能集成了进去，这个就见仁见智了，并不是谁都认为代码测试是日常开发工作的必要一环。

在我看来，这是学习和比较开发工具的过程中最有趣的一部分，甚至像“该选编辑器还是IDE”这样的问题，都能在这里找到答案。

说完“下限”，我们再接着来说说编辑器的“上限”。如果编辑器支持用户写插件来定制功能，那它的“上限”就很高了。像 Emacs、Eclipse 和 Atom 这样允许修改任意功能的工具来说，它们的“上限”理论上是无限的。但扩展编辑器这个事情，由于最终还是要落实到社区和个人，真正能达到的高度就很难说了。

Eclipse 是最流行的 Java IDE 之一，插件开发的语言也是 Java ，所以它的插件社区也是非常火爆。但是 Eclipse 把插件运行在主进程中，一旦插件的性能堪忧，就会影响到 Eclipse 本身的体验。

Atom 和 VS Code 的插件开发语言都是 JavaScript，可以说写扩展对于很多人是没有门槛的，因为大家多少都会写一些 JavaScript。从这个角度看，对于 Atom 和 VS Code 这类基于浏览器和 JavaScript 引擎的编辑器，插件社区的活跃度是不用担心的，主要还是要考虑如何避免跳进 Eclipse 的坑里。

而Emacs 的插件语言是 Lisp 方言，Vim 则是使用自己的VimL，语言相对小众一些，对于很多人而言成本就高了一截。

上面我说了很多关于编辑器“上限”和“下限”，其实这也就是我学习新编辑器时会着重研究的方向。在研究时，其实我有个“三步走”的演变过程，也就是极客时间团队提到的学习路径：

- 最开始的时候，我特别关心快捷键和语言支持，在这上面花了很多时间，这个过程就像是打怪升级，不断更新自己的装备库；
- 再往后，我就会开始挑剔编辑器的其他组件，但凡是跟自己的工作习惯或者工作流不匹配的，就会想办法换掉它，这是个做减法的过程；
- 最后一步，就是自己学习写插件了，编辑器本身的功能和社区不能够完全满足自己的需求，本着“麻烦别人不如磨炼自己”的精神，我开始自己动手。

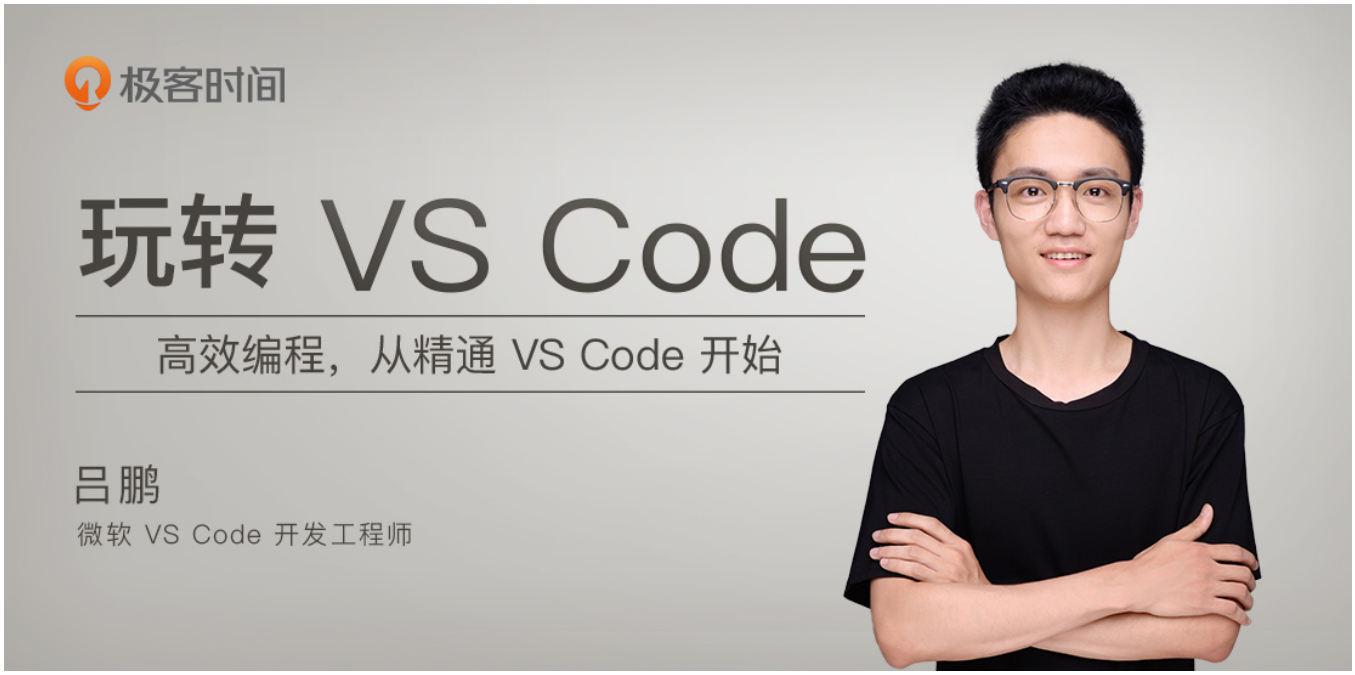
当然，一个人的兴趣点和精力都是有限的，任何事都没办法一蹴而就，因此有的放矢才是精进的关键。

我的建议是这样的，快捷键的学习是值得多花时间的；语言的支持方面，如果你是初学者，我不太建议使用一款什么都得自己配置的编辑器，把时间放在这上面不太划算（通常来说，语言或者框架作者推荐啥我就先用啥，等经验丰富了，再去研究那些更灵活的编辑器，才更游刃有余）；至于工作流这块，选择“有态度的”编辑器可能会更好一些，一款流行且社区活跃的编辑器推崇的工作流，大概率上是经得起工程上的考验的。

而插件开发的学习，既可以当做语言的学习，也能够当做业余项目，还能够锻炼跟社区的沟通能力，每个人都值得尝试。我在开发 VS Code 和一些插件的时候，就认识了不少朋友，这些人都非常优秀，也一直激励和影响着现在的我。

以上就是我在学习编辑器时的一些感悟，以及对于怎么学习编辑器、该学什么的一点看法，希望对你的编辑器探索之路有所帮助。当然，学习没有捷径，最重要的永远是立刻动手。

最后，也欢迎你在留言区和我分享你的编辑器学习经验，也许你会比我更高明。



精选留言

不做键盘侠
meta键是什么键?
2018-09-13 15:17

Ethan_zyc
VS Code 适合开发 Java 吗，现在还是乖乖用的 JetBrains 的 IDEA
2018-09-13 11:58

A2020
学习新的编辑器，优先了解它的“下限”和“上限”，了解过程分三步走。
第一步：了解编辑器的快捷键和语言支持，快捷键值得多花时间；
第二步：开始挑剔编辑器的其他组件，但凡是跟自己的工作习惯或者工作流不匹配的，就会想办法换掉它，这是个做减法的过程；
第三步：最后一步，就是自己学习写插件了，编辑器本身的功能和社区不能够完全满足自己的需求，本着“麻烦别人不如磨炼自己”的精神，我开始自己动手。

“三步走”的演变过程，作为一个通用的学习新工具的方法是我在本篇文章中的最大收获。
2018-09-14 09:41

1900
VS Code适合开发Go么？若适合，希望老师在以后的课程中可以详细讲解下相关配置，很期待后面的课程。
2018-09-13 09:51

作者回复

VSCode 是谷歌 Go 团队官方推荐的编辑器，这个你放心使用好了

2018-09-13 10:51

Nevea
关于下限 VScode的自动补全貌似不如Atom，shell就没有补全 写个if for都要自己打出来 安装了一个shell Snippet也不理想，是不是VScode没有关注shell啊 或者我哪里没有设置好
2018-09-13 15:40

fisher

vim不是用lisp做扩展吧?

2018-09-13 07:53

作者回复

谢谢指正，Vim 使用自己的 Viml 不是 lisp dialect

2018-09-13 10:54

A_Elite

meta键是mac上的，windows是win键

2018-09-14 21:15

Moorez

老师，vscode编辑器有时候很卡是什么原因 就是文件数多的时候，会不会是插件造成卡得原因，换成 webstorm 就不会卡了？您有没有遇到过这样的情况?

2018-09-14 13:06

作者回复

是打字都会卡吗？那就问题大了，欢迎提供更多细节

2018-09-18 11:08

李思阳

想问下老师，使用vscode进行c++和shell的系统开发合适吗?

2018-09-14 06:07

昂头的笑脸

如果vscode很好的支持vim，可以考虑切换

2018-09-13 13:43

作者回复

有几个 Vim 的插件，我是其中一个的 maintainer ，这几个插件各有千秋

2018-09-18 11:12

大鹏

吕老师，课程中提到的vscode的插件开发受编程语言限制吗？是否需要提前具备typescript基础?

2018-09-13 11:17

作者回复

需要 JavaScript 的知识储备

2018-09-18 11:13

一步

Vscode的快捷键可以json文件的格式快速导入导出吗?

2018-09-13 11:11

QS

VS code的快捷键还是很符合使用习惯的，上手很快，也不与系统冲突。在我用过的IDE和编辑器中，比较奇葩的快捷键设置应该是jetBrains系列IDE：默认设置下，Ctrl + Y键是删除行，而不是恢复上一步操作；不熟悉的情况下就容易造成误删（幸好Ctrl + Z是正常的），所以我觉得在快捷键设置上下功夫确实很重要。

2018-09-13 01:49

作者回复

只要能够允许修改快捷键绑定，就还是可以的

2018-09-13 08:54

鹏
学习了
2018-10-01 20:11

张柏林
划重点
2018-10-01 19:11

夏日朝阳
老师的方法论写的很好，受教了
2018-09-17 11:38

忘詞。
能出个教程用VSCode调试Golang就最好了
2018-09-15 16:20

谢mingmin
在写一个串口扩展插件，很期待后面的文章😁
2018-09-14 12:43

mcintype□
右边编辑区选中文件时，左边的文件树会Focus到当前编辑的文件，无论是vscode还是基于vscode的其它编辑器都这样！文件树这边跳来跳去展开文件夹带来烦恼，eclipse是可以做到的，vscode如何设置?
2018-09-14 07:54

作者回复

你需要的是 explorer.autoReveal

2018-09-18 11:12

Senhor
我居然还没能用它启动过一个java项目，需要配置的东西有点多
2018-09-14 01:50