# 关于Python编程的一些问答

## 导语

大约1个月前，oschina.net和华章图书一起合作做了一个活动：[OSC第51期高手问答——聊聊python那些事](http://www.oschina.net/question/1765084_162485" \t "_blank)，来推广我参与撰写的书《编写高质量代码：改善Python程序的91个建议》（[豆瓣链接](http://book.douban.com/subject/25910544/" \t "_blank)）。在回答问题的过程中，我看到有若干问题是好几个人都问了的，就萌发了在事后把这些问答整理整理的想法，以下内容就是来自那一次的问答。为简化整理，已经去掉了提问人的昵称，并做了简单的分类。

## 纠结的Py2与Py3之选

Q:Python 3 会导致Python的什么前景？最终由3统一，还是一直分裂？各自用自己的版本？

A:py3自发布以来，进步很大，解决了很多py2无法解决的问题，所以我是坚信大家最终会转到py3。现在py3已经有许多非常吸引人的特性，比如yield from，比如asyncio，比如更漂亮的库结构，等等。但我承认py3还没有一个巨大的吸引让大家转过去，yield from 算半个，另外半个我觉得是jit，想像一下py3自带jit，运行速度是py2的3到10倍，大家肯定一窝蜂转过去了。我觉得py3是未来，但也赞同两个割裂的版本影响推广。

Q:新手入门应该学2还是3？2会被Python团队放弃吗？

A:学py2吧，如果到时要转py3也是很容易的事。但如果直接学py3，到时候项目要用py2，就会觉得由奢入俭，很痛苦。

Q:请问您觉得Python3.x需要多久才能成为主流？

A:还有比较长的时间，但如果py3.5有独占的、可靠的、官方的jit方案的话，应该会加速很多很多！

## 怎样学习Python

Q:Python适合作为一个编程入门语言吗。

A:考虑到就业等，我觉得C语言还是更适合作为入门。

Q:python学习的进阶？基本知识掌握后，该怎么学习？

A:如果已经在工作了，那就直接尝试用python去解决工作需求就好了；如果是学生，那就去复制已存在的网站，把它的功能都弄来，比如oschina。

Q:高质量的代码是怎么写的，怎么提升自己的代码的质量？

A:个人看法：1、熟悉语言的细节；2、熟悉语言和库的最佳实践；3、多看一些提升代码质量的指导书籍；4、同行评审；5、多学几门语言，博采众长。

Q:对一个java开发者来说学习使用python有什么好的建议吗？

A:像当年学习java一样学习它。

Q:初学者，有什么好的网站推荐学习？

A:还是读书、看手册吧，网站的知识太零散，不成体系，容易学成野路子程序员。

Q:学习和使用了一段python后怎么做才能更好的提高自己使用python的能力呢？

A:我的回答是复刻一些产品，比如自己尝试做个豆瓣、oschina之类。

Q:《编写高质量代码：改善Python程序的91个建议》这本书适合其它非Python程序员看吗？

A:不适合。它的定位是对python有所了解的人。给初中级python程序员提升到中高级，这样的定位。

Q:请推荐几个比较优秀的Python开源项目，用来学习的。

A:优秀的python开源项目啊，我想一下，额，看一下trac和reviewboard？

Q:你觉得你学Python以来觉得关于Python最好的书籍有哪些呢？语言基础、语言设计、框架方面，等等都行。谢谢了。

A:很好的问题，我觉得可以回答，因为我自己觉得自己的书也不是“最好的”那个层次。推荐《expert python programming（中译Python高级编程，翻译差强人意）》、《Python源码剖析》。

Q:你好，我有个问题想请教你，如何成为真正的pythoner。我以前是做C++的，现在工作会接触一些python web方面的任务。我最开始接触python的时候学了些基本的语法就开始尝试写与算法相关的程序了，所以写python程序的时候代码还是有C++的影子，直到现在工作中维护之前别人的python web程序的时候，发现自己的python程序太不优雅了，好多时候有的功能可以用更美更快更高效的方式实现，但我可能会像C++里面一样想要从底层一点一草一木的搭建、控制我的程序，后来发现可以几行甚至更短的代码就可以实现那些功能，而且执行效率也更好一点，于是发现写python程序不能用以前C++代码时的思想，或许我可以尝试更pythoner一点？

A:很有意思的一个问题，个人建议如下：1、通读一遍手册，特别是lib ref和lang ref，所谓熟能生巧，对语言本身的熟悉才能写得更pythonic；2、多看一下经典的python项目的文档，比如flask，比如pip，看看他们提供了什么机制，甚至探究到这样的机制是怎么提供的；3、不要担心，多写一些python代码就好了。

## 应用Python的困惑

Q:Python 适合开发比较复杂的web项目吗？

A:在我看来，python适合开发小中大巨多种复杂程度的项目，因为能不能把项目做成，最关键的因素还是人。python对web支持不错，有许多第三方库，也有django/flask等许多人叫好的框架。

Q:入门Python用哪个操作系统环境好？我就只在Windows上学过一点基础，但不知一般真实团队开发环境是怎样，可以可以介绍下？谢谢！

A:我们一直用linux作为生产环境，mac os x 是我的开发环境，我用过许多年的windows，我不觉得使用windows会影响你的学习。

Q:目前python似乎大多数都是在WEB方向的应用，对于做C语言和C++的开发python有什么可以帮助的地方

A:主要是利用动态语言的灵活性、解释型语言的方便性，来解决C/C++在应变多变的业务需求、快速部署等方面的成本过高或力不从心的问题。

## 经验之谈

Q:学生，只往PyQt方向发展可行吗？

A:从长远来说，一里通百里融，只要有个突破口，后面有成就也是很可能的。从短期来说，感觉这方面就业面比较窄。

Q:感觉python什么都可以做，但是很难做到很好。如果没有c/c++/go扩展， python能做出大吞吐量，高并发高稳定性的系统吗

A:说python什么都可以做，还是高估了，但说python很难做到很好，就得看“很好”这个词怎么定义了，估计各人的标准还不一样。恕我见识少，好像用go来扩展python还没有成熟方案？目前我经历的网游、网站方面来说，我觉得还可以做出大吞吐量的高并发系统的，只是可能硬件成本会稍高些，至于稳定性，决定性的因素是开发人员的质量，跟语言关系不大，甚至c/c++更容易开发出不稳定的系统。

Q:python 的类库，函数库庞大，如何能快速找到自己需要的类库？

A:这是一个问题，很多语言都有同样的问题。要不你先去看一下 awesome-python 这个项目？

Q:python程序员收入咋样

A:好像年收入从几万到几十万的都见过，但上百万的我还没有见过。总得来说，我觉得跟其它语言的差不多，但高薪的比例可能是比较靠前的。

Q:我原是一名ruby程序员，后来看到python有非常多的模块，如ipython、ipython notebook等非常酷的python应用，其实在灵活性上python远不如ruby（method\_missing等），设计哲学也不一样，想问一下为什么python比ruby的应用多很多？特别是科学计算（Numpy）和绘图（matplot）？

A:于python会在科学界这么流行的原因，据说是因为那些科学家都不是计算机专业的，觉得python这货容易学容易用，所以就用这个，反正只是一次性地写写脚本跑一下试验数据。

Q:我想请问，python用来写游戏好么，大家不都说python相对运行起来比c++/c慢，而现在大部分游戏引擎也都是c/c++,就看那个Cocos2d，最先貌似是python版，之后还是用c/c++重写，虽说现在Python版也有更新；另外，现在移动端的各种软件游戏开发也比较热门，python在这方面好像有点跟不上的感觉。

A:是，在手游时代，python已经不合适编写客户端了。可以尝试在服务器端使用它，还是很好的。

Q:我有这么几个问题想请教一下：1. 多框架、多模块的实现语言代表着多学习成本吗2. 胶水语言的存在，现在主要用来做什么，其他层面上是如何应用的3. 类 Unix 系统管理，如何学习他们4. Web 方面与 ruby 之类的有哪些区别，或者说，优缺点5. 写过文字性的爬虫，感觉字符集和平台差异稍有异样，请问您是如何避免的

A:1. 我不太明白多框架、多模块是什么意思，我可以理解为python有很多框架吗？如果是的话，我觉得并不代表更多的学习成本，你看一下quick-start，看一下examples，看一下doc里关于扩展和分拆的机制，就知道这个框架、模块是否适合你的技术观，适合的就行，不适合的就不学，不学不会有成本；2. 胶水语言的这个问题，我觉得《unix编程艺术》这本书里谈得比我讲要好得多，推荐看； 3. 关于sa这一块，我了解不多，建议查阅专著；4. ruby的好处就是有ror这样的大一统解决方案，python是选择多，麻烦也多；4. 文本处理，在编码上的问题很多、很难，建议使用 chardet 等方案，但都是有力不能及的地方的，建议接受部分问题是无法解决的现实。

Q:以下的几个问题想请教一下的：(1) 如何做到高效的python coder, 我也是近这2年才开始用python的，但觉得开化效率还有很多改善的地方，与相对公司内部的java组来说。(2) python 的其中一点我比较喜欢的是比较明了，什么东西都比较原生态，相当于.net来说过于包装，这就是它的长处，如果现成的lib也可以找到，请问一下平时一般常用的东西，是自己做成lib好，还是直接用别人写好的lib好？谢谢(3) python是否比较适合写web呢或是比较好的选择？相对于国内这个环境来说，python这一方面真的比较少些，当然国外有google做带头大哥，它期下的很多projects都很出彩的，我现在做一个web 方面的project，选择bottle，它和flask比较类似，速度方面会较好一点，发现项目进行中遇到不少关于技术方面的问题，可查找到的资料比较少，解决问题相当困难。

A:1. 怎么做高效的python coder，我觉得跟其它语言没有二致，不管怎么样，多读官方文档肯定大有增益，这也是我的经验之谈。2. 我一般直接用别人写好的lib，如果有而且用起来爽的话；3. 建议选择flask/django等社区比较大、比较活跃的框架。

## 编码一线

Q:个人感觉python没有成熟的IDE，写程序都是文档不停不停地翻，不想写C++或用eclipse，都是自动提示+文档，我想知道是我实在是才疏学浅还是python开发的通病呢？

A:嗯，其实没有特别厉害的IDE，pyCharm是比较好的，而且有免费的社区版，习惯了用E记的，可以试一下pydev。

Q:想问一下，python每import一个模块都会在内存中实例化还是共享一个实例。

A:默认是共享一个。

Q:您是用什么工具打包python工程的？我使用pyinstaller打包包含gtk的程序后，执行打包后的文件会报\_glib module 不存在。不知道还有什么更好的工具没。最好是跨平台的。

A:我们一般不打安装包，我们一般就是用setuptools。

Q:python“函数”的返回值类型不在语句的语法声明中，使用python开发项目，由于返回值类型不那么明确，怎么确保在软件迭代中不会弄错？

A:一方面是大家遵守一些最佳实践，比如保证返回值都是同一类型的；二是充分进行单元测试；三是使用较新的python版本，已经支持参数和返回值的类型声明。

Q:对于python 的协程 有什么好的库比较好用

A:必须是gevent。

Q:请问Python有类似Ruby社区RVM、Bundler、Rake的成熟工具链么？

A:有的，pypi.python.org，你可以上去看一下。我书里也有提到这些工具链的用法。