

认识 Python (转自互联网)

一、Python 的简介

正是因为应用开发工程师、运维工程师、数据科学家都喜欢 Python，才使得 Python 成为大数据系统的全栈式开发语言。

对于开发工程师而言，Python 的优雅和简洁无疑是最大的吸引力，在 Python 交互式环境中，执行 `import this`，读一读 Python 之禅，你就明白 Python 为什么如此吸引人。Python 社区一直非常有活力，和 NodeJS 社区软件包爆炸式增长不同，Python 的软件包增长速度一直比较稳定，同时软件包的质量也相对较高。有很多人诟病 Python 对于空格的要求过于苛刻，但正是因为这个要求，才使得 Python 在做大型项目时比其他语言有优势。OpenStack 项目总共超过 200 万行代码，证明了这一点。

二、Python 的优势

对于运维工程师而言，Python 的最大优势在于，几乎所有 Linux 发行版都内置了 Python 解释器。Shell 虽然功能强大，但毕竟语法不够优雅，写比较复杂的任务会很痛苦。用 Python 替代 Shell，做一些复杂的任务，对运维人员来说，是一次解放。

对于数据科学家而言，Python 简单又不失强大。和 C/C++ 相比，不用做很多的底层工作，可以快速进行模型验证；和 Java 相比，Python 语法简洁，表达能力强，同样的工作只需要 1/3 代码；和 Matlab，Octave 相比，Python 的工程成熟度更高。不止一个编程大牛表达过，Python 是最适合作为大学计算机科学编程课程使用的语言——MIT 的计算机入门课程就是使用的 Python——因为 Python 能够让人学到编程最重要的东西——如何解决问题。

三、为什么要学 Python

经常有人在群里问，运维人员需不需要学开发？需不需要学 PYTHON ？ PYTHON 和 SHELL 有什么区别？

现阶段，掌握一门开发语言已经成为高级运维工程师的必备技能，不会开发，你就不能充分理解你们系统的业务流程，你就不能帮助调试、优化开发人员开发的程序，开发人员有的时候很少关注性能的问题，这些问题就得运维人员来做，一个业务上线了，导致 CPU 使用过高，内存占用过大，如果你不会开发，你可能只能查到进程级别，也就是哪个进程占用这么多，然后呢？然后就交给开发人员处理了，这样咋体现你的价值？

另外，大一点的公司，服务器都上几百，上千，甚至数万台，这种情况下怎样做自动化运维？用 SHELL 写脚本 FOR 循环？呵呵，歇了吧，SHELL 也就适合简单的系统管理工作。到复杂的自动化任务还得要用专门的开发语言。

你可能说了，自动化管理有专门的开源软件\监控也有，直接拿来用下就好了，但是现有的开源软件如 puppet\saltstack\zabbix\nagios 多为通用的软件，不可能完全适用你公司的所有需求，当你需要做定制、做二次开发的时候，你咋办？找开发部门？开发部门不懂运维的实际业务逻辑，写出来的东西烂烂不能用，这活最后还得交给运维开发人员来做。

其次，不会运维开发，你就不能自己写运维平台\复杂的运维工具，一切要借助于找一些开源软件拼拼凑凑，如果是这样，那就请不要抱怨你的工资低，你的工作不受重视了。

PYTHON 第一是个非常牛 B 的脚本语言，能满足绝大部分自动化运维的需求，又能做后端 C/S 架构，又能用 WEB 框架快速开发出高大上的 WEB 界面，只有当你自己有能力做出一套运维自动化系统的时候，你的价值才体现出来，你才有资格跟老板谈重视，否则，还是老老实实回去装机器吧。

四、Python 和其它语言的对比

运维开发为什么要用 Python ？

Good question, 为什么不用 PHP , JAVA , C++ , RUBY , 这里只能说, 见人见智, 如果你碰巧已经掌握了除 Python 之外的其它语言, 那你爱用啥用啥, 如果你是一个连 SHELL 都还没写明白的新手, 想学个语言的话, 请用 Python , 为什么呢? 首先, PHP 是跟 PYTHON 比的最多的, 其实他俩根本就不用比, 为什么呢? 两个语言适用性不同, PHP 主要适用于 WEB 开发, 可以迅速的做出中小型, 轻量级的 WEB 网站, 但后端嘛, 基本还是要借助其它语言, 借助什么语言呢? SHELL ? Python ? 而 Python 呢, 是个综合语言, 前后端都可以, 单拿出来比 WEB , 也一点不比 PHP 差, 但为什么 WEB 方向上 PHP 比 Python 要火? 先入为主嘛, PHP 90 年代诞生就是做 WEB 的, Python 2000 年后才出现 WEB 框架, 但论优秀程度上, Python 的 WEB 框架基本上出其无左, 至少是跟 PHP 比。

那 JAVA 呢? 好吧, 一个臃肿\中庸\毫无新意的语言, 还是老老实实用它来做 ERP 吧, 搞个运维小平台, 用 JAVA 真心没啥必要, 在我看来, JAVA 就是稳定的中年男人, 稳定\成熟\秃顶, 而 PYTHON 代表的就是青春, 简洁\快\干净\帅!

C++ \ C , 这个嘛, 我只能说, 如果你会了 Python, 又会 C 的话, 那你会更吃香, 但是不会 C 的话, 其实也无大碍, 基本上做运维的人, 搞搞 C 就是为了来装 B 的, 因为多数情况下你都到不了看系统底层源码的程度。

RUBY , 小日本开发的, 还不错, 风格跟 Python 有点像, 因为 ruby onrails 出了名, 国外用的比较多, 国内, 放心吧, 没戏, Python 已经把它想象空间都占死了。

当然还有新的语言 GO , 有些搞运维的看见做开发的人员搞 GO , 也想凑热闹, 觉得是未来, 我想说, 别瞎没事跟风, GO 再成功, 也不会变成运维开发语言。

五、认知自己，提升技能

有些人觉得 Python 效率低,说他不能支持多线程, OH ,好吧,这个还有点说对了,但是我想问,看我这篇文章的有几个做过搜索引擎开发?有几个做个上亿 PV 的并发网站开发?有几个看过 LINUX 内核源码?如果没有,请别瞎跟着传了,知道 PYTHON 为什么不支持多线程吗?这句话问错了,其实 Python 支持多线程,只是不支持多 CPU 多线程,也就是一个程序 spawn 出来的多线程只能占用一个 CPU ,但是为什么呢?噢,因为 GIL , GIL 是什么东东,请自行补脑。。但是你非得用多线程吗?你可以用多进程呀,再牛 B 你还可以用协程呀,这些 Python 支持的都很好呀,如果你的程序逻辑不好,搞个多线程也快不起来。

我认识一个博士讲过一句话,我觉得不错,他说,程序效率高低, 80% 都是写程序的人决定了,语言本身就占 20% ,所以下次有人再说 PYTHON 效率低的时候,请让他先回去自己检查下自己的程序多了多少无用的逻辑、循环等等。这个博士自己用 PYTHON 写的 WEB 程序,一台服务器每天能处理上亿请求,一秒并发近两万,什么 WEB 框架这么牛 B? 别问它是谁,它叫 tornado 。

六、Python 能否自学?

当然可以,什么都可以自学,前提是你得能学得会,见过 N 多菜鸟踏上上自学的不归路,他妈的什么都能自学的话,还用大学干什么?自己在家整不就行了?动不动就说 Python 是个脚本语言,自己看看就不会了,说这话的只可能有两种人,一种是高手,一种是 SB ,对于高手来讲,他肯定已经会其它语言, Python 在这种情况下,自学当然就很容易学会,几年前我刚接触 Python 时,代码遇到问题,找了个开发的哥们帮调试,哥们帮调了十几分钟就搞定了,结果人家以前一句 Python 代码也没写过,为什么,因为语言都有相通之处,一门掌握好了,其它门自己学学就会了。