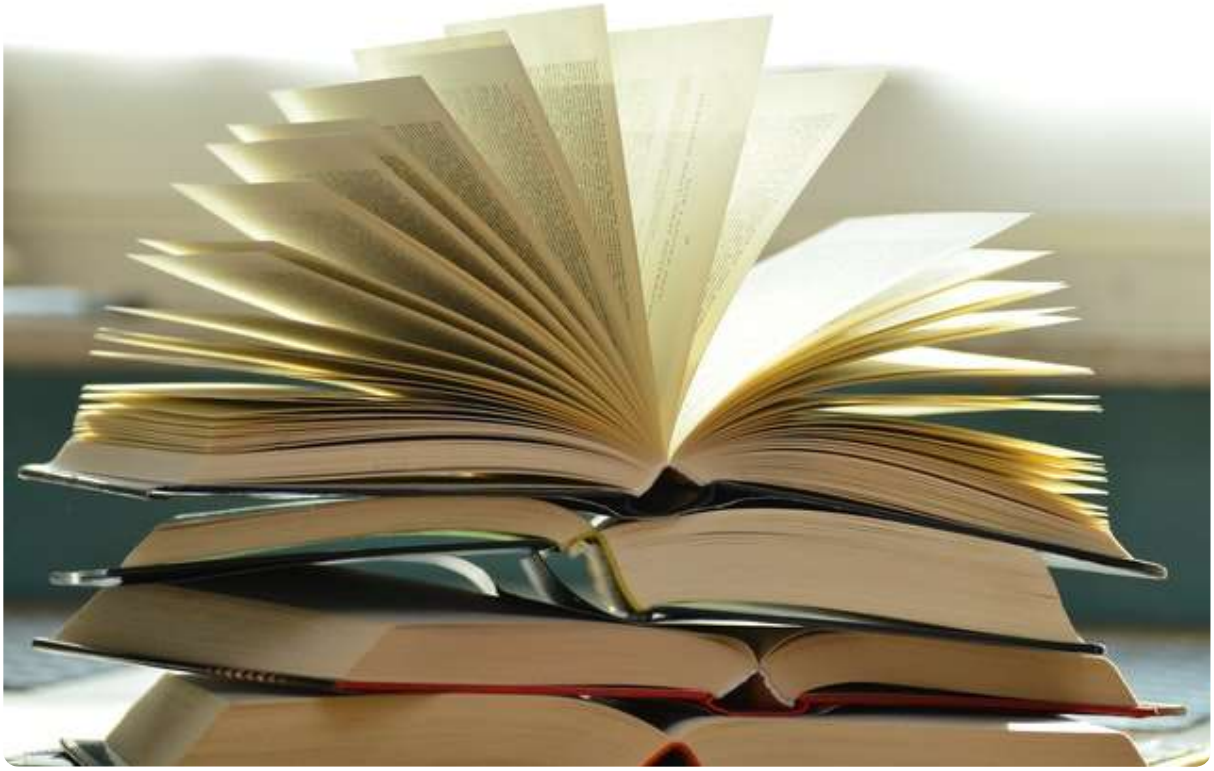
01 开篇词：为什么要钻研 Python 源码?

更新时间：2020-06-19 11:20:47



> 没有引发任何行动的思想都不是思想，而是梦想。
>
> —— 马丁

为什么要钻研 Python 源码

大家好，我是 *fasionchan* ，资深 *Python* 工程师，曾就职于 **腾讯** 、**网易游戏** 、**蚂蚁金服** 等互联网公司。

经过多年的摸爬滚打，我对 *Python* 开发有一些自己的体会，深知源码研究的意义。我就职于网易游戏时，曾在内部开展 **Python源码剖析** 系列技术分享，收获颇多。

而不知其所以然，因而与工作机会无缘，非常可惜。

随着人工智能等技术的兴起，*Python* 正风靡世界！最新的 *TIOBE* 编程语言排行榜上，*Python* 已经稳居前 *3* 位！正如硬币的两面，这对 *Python* 工程师来说既意味着 **机遇**，也带来一些 **挑战**。

排名	编程语言	流行度	对比上月	年度明星语言
1	Java	17.358%	^ 0.462%	2015, 2005
2	C	16.766%	^ 0.993%	2017, 2008, 2019
3	Python	9.345%	v 0.359%	2010, 2007, 2018
4	C++	6.164%	^ 0.59%	2003
5	C#	5.927%	^ 0.578%	
6	Visual Basic .NET	5.862%	^ 0.575%	
7	JavaScript	2.060%	v 0.391%	2014
8	PHP	2.018%	v 0.387%	2004
9	SQL	1.526%	^ 0.022%	
10	Swift	1.460%	v 0.335%	

一方面，采用 *Python* 技术栈的公司越来越多，掌握 *Python* 的工程师不愁没有工作机会。

另一方面，大量的工程师浩浩荡荡加入 *Python* 大军，竞争日益激烈。如果你想要从众人中脱颖而出，就必须拿出自己的看家本领。

不少初学者觉得，能用 *Python* 完成开发需求就行了，没必要深入学习。这个观点是非常错误的，掌握语法只是万里长征第一步。

程序能跑起来，并不意味着程序能高效地运行，实际上这两者有天壤之别。制约程序效率的因素很多，语言特性、运行机制、算法原理、操作系统环境等等，只掌握皮毛显然无法驾驭这些。

归根到底，还是在底层原理上吃亏。不管学习什么技术，浅尝辄止肯定是不行的。因此，要么满足于 *Python* 语法，沦为 *API* 调用侠；要么，彻底掌握 *Python* 的内部原理。

研究过源码的我，深知这其中的艰辛——成千上万行代码，多少个日日夜夜以及笔记本里的一个个草图……如果能够将心得分享出来，相信对后来者会有所帮助。因此，当慕课网的老师跟我沟通时，我们一拍即合。

那么，这个专栏都介绍些什么内容呢?

第一部分， **开篇** 。讨论源码学习对提升开发能力的重要意义，结合“小菜”的经历介绍 *Python* 工程师的成长历程。

第二部分， **对象模型** 。介绍 *Python* 面向对象理论体系，明确“ **类** ”和“ **对象** ”在 *Python* 中的表现形式。通过源码，分析对象的 **内存布局** ，研究对象的 **生命周期** ，初步揭开对象的面纱。

第三部分， **内建对象** 。涵盖 *float* 、*int* 、 *bytes* 、*str* 、*list* 、*dict* 等内建对象的实现细节，重点突出每种对象的 **数据结构** 以及背后的 **算法思想** ，并结合 **工程实践** 讲解内建对象的 **妙用** 。

第四部分， **虚拟机** 。先介绍 *Python* 程序的执行过程以及 **字节码** 的结构；然后介绍 *Python* **虚拟机** 的运行机制，以及 **名字空间** 等运行时上下文；最后讲解 *GIL* **全局锁** 对虚拟机的影响并探索应对之策。

第五部分， **函数机制** 。探索函数从 **代码** 转化成函数 **对象** 的所有步骤，并以 *Python* 代码模仿这个过程。重点讲解 **嵌套函数** 、 **闭包** 以及 **装饰器** 这些面试必问概念，并结合工程实践介绍 **函数式编程** 和 **装饰器** 的高级用法。

第六部分， **类机制** 。探索类从 **代码** 转化为 **类** 对象进而创建 **实例** 对象的所有步骤，并以 *Python* 代码进行模拟。重点讲解 **继承机制** 、 **属性查找** 等高频面试知识点，以及 **魔术方法** 、 **元类** 在程序开发中的妙用。

第七部分， **生成器与协程** 。讨论 **生成器** 的高级用法，并以 **字节码** 讲解它的运行原理。基于 **生成器** ，动手设计一个协程库，加深对协程运行机制的理解。结合工程实践，讲解用 *asyncio* 构建 **高并发应用** 的技巧。

第八部分， **内存管理机制** 。结合 *Python* 特色全面讲解 **内存池** 、 **引用计数** 、 **标记删除** 、 **分代回收** 等关键技术。理论联系实际，以一个真实案例介绍工程实践中排查并解决 **内存泄露** 问题

第九部分，**总结**。回顾专栏内容，在读懂 *Python* 基础上，追求有 *Python* **格调**的代码设计技巧。

那么，学习本专栏学要具备哪些基础知识呢?

基本的 *Python* 开发能力是必须的，不需要特别深入，能独立编写程序即可。有一定的 *C* 语言基础更好，没有也不打紧。专栏从最简单的源码入手，逐步深入，并在必要地方补充介绍相关 *C* 语言知识。大家不必恐惧 C 语言，专栏精心编排，并不拘泥于繁缛的代码细节，总体通俗易懂。

此外，专栏附带大量的图表，详尽地描绘了 *Python* 内部各种数据结构间的关系。例如，*list* 对象的内部结构图：

相信在这些图表的帮助下，你可以轻松地理解原本很深奥的源码。虽然绘制这些图表花了我大量的时间，但这一切都是值得的。希望我一个人的时间付出，能帮更多的人节约学习时间，起到事半功倍的效果。诚如是，则不胜荣幸!

著名技术作家侯捷在《STL源码剖析》中说过：源码之前，了无秘密。源码就像武功秘籍，虽然有时难懂如天书；可一旦参透，便掌握绝世武功的奥妙！感恩学生时代花了无数日夜苦读《STL源码剖析》的自己!

源码之前，了无秘密！与诸君共勉!

02 小菜成长之路，警惕沦为 API 调用侠 →

欢迎在这里发表留言，作者筛选后可公开显示

星星在线

我简直要泪目了，我一直在等Python源码剖析那本书的新版本问世，已经望眼欲穿的时候。这个专栏，它就这么突然的、猝不及防的出现了，我简直要起飞了。以前做C++的时候最喜欢的就是《STL源码剖析》和《深入探索C++对象模型》这两本书了，一直感叹Python为什么就没有这种书，终于让我看到它出现的契机了，感谢作者

0 回复 11小时前

weixin_Alfa_0

我一直觉得，Python的解释器是用C语言来实现的，这件事情非常有趣，语言可以诞生新的语言。学习一门语言，原来只在乎语法特性和源码，现在看来，语言是如何实现的，理解了这个，就抓住了本质，语法只是表象。

2 回复 2天前

fasionchan 回复 weixin_Alfa_0

同感~

回复 1天前

jxs1211

python的能做微服务吗，求推荐的开源资源吗

0 回复 2020-07-15

fasionchan 回复 jxs1211

完全可以，这方面开源资源非常多：web接口服务可用flask、fastapi等；rpc服务可以用grpc；消息队列可以用rabbitmq、kafka等；还有一个异步任务框架celery；数据库有mysql、mongodb、redis等等，将这些技术组合起来可以实现相当大规模的应用。

回复 6天前

fasionchan 回复 jxs1211

技术选型需要针对具体场景考虑，Python的长处在于强大的表达能力，完整的技术生态和极高的开发效率，短板是性能上有瓶颈。如果你的业务规模不是特别海量，对性能不是特别敏感，Py

回复 6天前

血小板自动机

老师使用的是python3的源码来进行剖析吗

2 回复 2020-06-13

fasionchan 回复 **血小板自动机**

是的。由于官方已经停止对Python 2进行维护，未来是Python 3的世界，因此专栏以Python 3源码为目标讲解、剖析。

回复 2020-06-15 10:32:52

哈煌

fasionchan老师，感觉底层的原理都是计算机专业学生在大学的上的专业基础课，如《计算机操作系统》《计算机组成原理》《计算机网络》《数据结构》等等，只是大学学的是理论，实践得少。工作后，是在实践中与这些原理进行对应，具体问题具体分析，背后的原理还是那些基础课程。所以，我觉得，相关从业人员有必要去重新回顾和学习这些课程的。欢迎讨论和交流哈。 看了开篇，给我的感觉是，这个专栏给可以给我们提供一个理论和实践的指导，我们之后学习其他编程语言，也可以参照老师的这个逻辑。很赞。 侯捷赞，让我重新认识C++。

4 回复 2020-05-18

fasionchan 回复 **哈煌**

你说得很到位，大学开设这些基础课程是有原因的。在我看来，《计算机组成》、《计算机网络》、《操作系统》、《数据结构》以及《数据库》这些基础理论，相当于于武侠小说中的内功。只有将内功基础打扎实，才能走得更远。可在工作之余抽空研究一番，假以时日，必有脱胎换骨的感觉。 美中不足的是，这些基础课程大多很理论化，学起来比较枯燥。如果可以将理论与具体的项目实践相结合，配以活生生的例子，并用更通俗的语言进行讲解，效果可能更好。这可以是你我，以及其他创作者们努力的方向。 侯捷是我很喜欢的一位技术作家，大学时代看过他写的很多书，获益良多。在国内技术写作界拼凑之风盛行的背景下，我特别钦佩他的治学态度。我虽能力不及前辈万一，也会以他为榜样，努力将这个专栏写好。欢迎各位亲多提提建议，帮我完成这个愿望。

回复 2020-05-19 10:29:50