



链滴

学习技术的三部曲：WHAT、HOW、WHY

作者：[c3gen](#)

原文链接：<https://ld246.com/article/1505705793011>

来源网站：[链滴](#)

许可协议：[署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

最近几天有些网友在邮件里面问我关于学习的问题。

有好几个人觉得工作了几年，也学会了不少的类库、框架、甚至语言，但是感觉自己的能力没有太大提高。因此今天来说一下我个人对这方面的体会，希望对大伙儿（尤其是新手）有帮助。

先声明一下，本帖子讨论的三部曲是指你已经选定了某个技术方向之后，该如何学习；至于如何选定技术方向，则属于另一个话题，不在今天的讨论之列。

我把学习归类为三个步骤：What、How、Why。

经过我对周围同事和朋友的观察，大部分感觉自己技术没有提高的人，都仅仅停留在What阶段。

下面我把这三个步骤解释一下。

★★★★★第一步：WHAT

“WHAT”也就是“What is it?”——这是最简单的层次。在这个层次，你要搞清楚某个东东是什么子的？有什么用？有什么功能特性？有什么语法？.....举例如下：对于学习语言（比如C++、Java、Python），大部分人都能够掌握基本的语法和标准库，然后用它写一些小程序（诸如二分查找、冒泡排、简单文件操作等）。对于学习类库（比如JDBC类库），大部分Java程序员都能明白JDBC主要包含些类，也能够用JDBC进行简单的数据库查询和增删改操作。由于这个步骤是最基本的，假如你连这做不到（可能你的理解力不够好），也别在IT界混了。但是光会What是不够的。仅仅停留在这个步，导致了很多程序员只知其然，不知其所以然。这就是目前大部分开发人员的现状。

★★★★★第二步：HOW

所谓的“HOW”就是“How to do?”。在这个层次，你要搞清楚某个东西，其内部是如何运作的？何实现的？.....举例如下：假如你在学习C++语言，你是否搞明白函数传参数的实现机制？虚函数是何实现？抛出异常时的栈回退是怎么回事？.....假如你在学习Java语言，你是否搞清楚GC如何实现？射是如何实现？.....假如你在学习JDBC库，你是否清楚JDBC Driver的4种类型？不同游标类型的实现制？事务的机制？.....在这个阶段，你必须多想想类似这些问题。然后通过各种途径（参见“关于自能力”的几个方法），把问题彻底搞清楚。自然而然，你的提高就会比较明显。而且如果碰到一些深层次的问题（比如性能优化），也就知道该如何去解决。完成这个阶段之后，你基本上就属于该技术领最优秀的20%的人（根据二八原理，80%的人不会去思考HOW的问题）。

★★★★★第三步：WHY

一般来说，只有想清楚HOW之后，才能继续去考虑WHY。所谓的“WHY”，就是搞清楚某个东西什么设计成这样？为什么不是另外的样子？这样的设计有什么讲究？.....说实在的，善于问“为什么有一定的天赋成分？好像某个科学大牛曾经说过“提出问题有时候比解决问题更难”。一般来说，只当你深刻理解了某个东西，才能够针对这个东东的设计问出一些问题。所以，我前面强调过，要先把OW的问题搞清楚，再来考虑WHY的问题。

举例如下：

- 1.对于C++语言：为什么C++没有类似Java的finally关键字？为什么C++当初没有考虑GC？.....
- 2.对于Java语言：为什么Java没有类似C++的类析构函数？为什么Java要同时提供String和StringBuffer两个似乎冗余的类？.....
- 3.对于Python语言：为什么Python不提供类似C++/Java的访问控制机制？.....如果你能够自己问出如上述的“为什么”问题，并且能够通过各种途径找到解答，那你基本上已经吃透这个技术了，并且已经有可能自己去设计一个类似的玩意儿了。

到这时，你已经踏上了通向技术高手的康庄大道。由于本博客偏重IT方面，所以今天举的这些例子多都是IT相关的，但是这个三部曲在IT之外的行业和领域，其实也能适用。如何举一反三，就看各位的性了。