

PROGRAMMER 新程序员

.004

21年技术见证 《新程序员》编辑部 编著



我们的技术时代 我们的程序人生

>>做技术、管理、创业、脱口秀、科幻……一书看尽技术生涯的若干种可能

>>C++、C#、TypeScript、MySQL、PostgreSQL、Vue.js创始人如何用技术改变世界



中国水利水电出版社
www.waterpub.com.cn

新程序员 .004

我们的技术时代，我们的程序人生

《新程序员》编辑部 编著



中国水利水电出版社
www.waterpub.com.cn

• 北京 •

卷首语：追随榜样，做最好的自己

每个人的价值与他感兴趣之事物的价值恰好相当。——美国著名学者Will Durant,《沉思录》代序

这是四十余位技术人跨越半个世纪，用脚走出来的一个个真实的程序人生故事！

为什么要做这个主题？回想当初策划，恰逢2022年春节，疫情下岁末年初的更迭，正是大家集体总结和思考的时候，讨论最热的三个话题：第一，什么是这一年或未来五到十年的大趋势？第二，怎么找到行业与技术捷径，快速成功？第三，我怎么过好自己程序人生？

这些问题，其实是个长期话题。提问的人，想找答案的人，都喜欢去看自己的偶像怎么答。尤其是技术圈，比武论英雄，只有自己认可的偶像所言，才值得信赖。

为了寻找答案，《新程序员004：我们的技术时代，我们的程序人生》邀请了四十多位程序员出身、在各自领域取得卓越成就的高手，以自己的阅历和洞见做回答。这些不同国家、不同领域、不同岗位，甚至不同年代的人物故事，汇集在一起，绘成这个时空维度下我们技术时代创建者的大致全貌。其间既能看到用技术改变世界背后的深邃思考，也能看到这些拥有着不同背景、兴趣的程序员们在不断的选择与挑战中，最终是如何找到自己的最佳道路。

第一，我的路在何方？雷军顺势而为趁着移动互联网之风创下小米；Bjarne Stroustrup说“C++的成功令人惊讶”，因为最初只是想改进C语言……八十年的计算机发展史，我们深刻地感受到了“小赢靠智，大赢靠势”。20年前，《程序员》试刊的封面文章《到美国去，挣美元！》，报道了共享软件MP3 CD Maker作者周奕走出国门，通过共享软件月入5万美元的故事。20年后，在程序员已经上可探“火星”、下可入“深海”的今天，人人都是开发者，家家都是技术公司，科技创新与开源开放已成为数字经济的引擎，我们也正迎来开发者技术大生态的黄金时代。那么，在技术细分领域下，什么是最好的时机？怎么做选择？且看技术大师与先锋们一路而来的真知灼见，前行者的光芒就是指路的明灯。

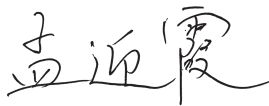
第二，快速成功的捷径是什么？这句话包含两点，首先什么算成功？其次才是成功的捷径。成功的定义多种多样，无论是用技术改变或造福世界，还是“只是为了好玩”，均可包含其中。天下武功，唯快不破，而快速成功，就需要抬头看路，在我们想要走的那条路上，是否有前人成功抵达的经验？这本四十多人的故事与经验，就是我们寻访海内外高手，获取到的真经秘籍。他们的思考、抉择都值得借鉴。

第三，怎么走出最适合自己的路？程序员从技术起航，乘风破浪：有人创造工具，成为技术大师；有人创造产品，成为热血创业者；有人掌舵企业技术的“诺亚方舟”，率万千舰队劈波斩浪；有人以梦为码，科技向善，为障碍人群开启互联网的大门；有人不囿于一隅，创造科幻世界的奇妙星球；有人不甘于沉闷，转型做脱口秀大咖，将程序员的可敬呆萌带出圈……

C#之父Anders Hejlsberg说：“编程是我的热爱！我一直致力于提高程序员的工作效率，这种爱与热情是我坚持的动力，我对这个行业的热爱无可阻挡”，“为这个世界贡献自己的力量，这也是每个人的理想，起码要在我们离开这个世界之前，给世界带来一点改变”。

每个人都在追求和实现自己的人生价值与使命，终其一生，技术是程序员的底层系统，支撑着我们去拓展自己的丰富应用与无限可能。榜样的力量，犹如黑夜中的一束光，总是吸引和激励着追随者们奋力前行。追随榜样，驾驭新技术，逐时代之浪潮，做出最适合自己的努力与选择，无关名利，无关身份，最终活成自己喜欢的模样。

世界因开发者而动，用技术的力量让世界更美好，这就是我们的初衷与愿望！



CSDN副总裁、《新程序员》总编

2022年5月



策划出品

CSDN

出品人

蒋涛

总编辑

孟迎霞

执行总编

唐小引

编辑

田玮靖 | 杨阳 | 屠敏 | 徐威龙 | 何苗 | 郑丽媛 | 郭露 | 宋林飞

特约编辑

罗景文 | 罗昭成

运营

张红月 | 杨过 | 武力 | 刘双双 | 白羽中

美术设计

纪明超 | 席傲然

读者服务部

胡红芳

读者邮箱: reader@csdn.net

地址: 北京市朝阳区酒仙桥路10号恒通国际商务园B8座2层, 100015

电话: 400-660-0108

微信号: csdnkefu



扫描二维码

观看更多精彩内容


专辑: 我们的技术时代, 我们的程序人生

- ① 卷首语: 追随榜样, 做最好的自己

技术名人堂

- ② C++之父Bjame Stroustrup: 程序员在数学上付出的努力永远不会白费 page.6
- ③ C#、TypeScript之父Anders Hejlsberg: 从事编程只因为热爱 page.10
- ④ MySQL之父Monty: 代码是一次写成的  page.16
- ⑤ PostgreSQL全球开发组联合创始人Bruce Momjian: “转行”是为了更好地前行  page.21
- ⑥ Vue创始人尤雨溪: 登榜GitHub之路看似不难  page.24

技术领导力

- ⑦ SUSE CTO Brent Schroeder: 开源、云原生浪潮来袭, 企业需做好准备  page.29
- ⑧ 阿里巴巴CTO程立: CTO就是要给CEO扫清障碍和风险 page.34
- ⑨ 百度CTO王海峰: 三十年旅程, 三次AI“翻译” page.43

技术先锋



- ⑩ 贾扬清: 追求大模型, 并不是一件坏事 page.50
- ⑪ 潘爱民: 程序人生三十年, 亲历计算机程序的演进 page.55
- ⑫ 阳振坤: 不做工程等于纸上谈兵 page.62
- ⑬ 章文嵩: 开源为我打开一扇窗  page.67
- ⑭ 云舒: 不畏“鸿鹄”之志, 只缘生于“安全”  page.74
- ⑮ 吴军: 不用低效率的算法做事情 page.81
- ⑯ 云风: 从“炫技”到追求简单, 40年游戏开发人生 page.87
- ⑰ 杨磊: 从数据平台20年发展, 看数据处理最优解 page.93

技术创业者

- ⑱ 戴志康: 技术创业, 从无到有, 从有到好 page.97

- ①9 邢山虎: 创业二十载, 选择远大于努力
page.101
- ②0 刘杨: 我的软件工程科研之路
page.105
- ②1 丛纹韶: 技术人“乱入”物流, 引发了哪些变革
page.110
- ②2 汤鹏: 从技术人到创业者, 我与时代碰撞出的那些“火花”
page.114

技术管理者


- ②3 钱岭: 别担心“35岁危机”, 要成为“老专家”
page.119
- ②4 潘娟: 90后女CTO的开源之路 
- ②5 陈恺: 清华、港中大学霸, GitHub“50000+”Star之路 
- ②6 李昊: 程序员如何完成从个人贡献者到技术管理者的职业转变
page.134
- ②7 邱良军: 数字化转型中, 我的研发管理心得
page.140

我是程序员

- ②8 祁宇: 我和现代C++的故事
page.146
- ②9 唐巧: 我的移动开发程序人生
page.150
- ③0 Phodal: 好奇心驱动的编程人生
page.154
- ③1 justjavac: 从辍学到Deno核心代码贡献者, 我的十年编程生涯
page.158
- ③2 蔡俊鸿: 30岁, 当一名游戏技术总监去
page.163
- ③3 周天涯: 从盛大、网易、电魂网络到字节跳动, 我的四次选择
page.169
- ③4 王禹华: 一名普通的女程序员, 没有“35岁危机”
page.172
- ③5 刘继聪: 从物理转AI、战数据库, 95后程序员的职业选择
page.174

女性力量



- ③6 “她”力量: 优秀AI女性共话成长与挑战 
page.179

- ③7 开源圆桌派: 开源世界里的女性力量 
page.184

技术向善

- ③8 黄刚: 我的信息无障碍技术之路
page.189
- ③9 刘彪: 黑夜给了我黑色的眼睛, 我却用技术寻找光明
page.192

斜杠程序员

- ④0 子寅: 不会“怼”产品经理, 干不了程序员, 更干不好脱口秀 
page.195
- ④1 白丁: 放弃编程、燃烧创意, 书写科幻人生 
page.199

百味

- ④2 《神秘的程序员们》之永恒的追求
page.203

本期配套视频内容总汇

- ① MySQL之父Monty: 代码应该是一次写成, 而不是后面再改
- ② Bruce Momjian: 35年数据库生涯, 我看到了这些变化
- ③ 尤雨溪: GitHub“顶流”背后的成长之路
- ④ CSDN副总裁邹欣对话SUSE CTO Brent Schroeder: 开源、云原生浪潮下的科技公司
- ⑤ 章文嵩谈入职淘宝的正确方式
- ⑥ 章文嵩: 从兴趣到职业, 30年编程给我带来了什么
- ⑦ 十年磨一剑: “欺骗防御布道者”云舒
- ⑧ 潘娟: 到达开源顶端的90后女CTO
- ⑨ 清华、港中大学霸, 如何成就GitHub 5万+Star开源项目
- ⑩ AI“她”力量: 四位杰出女性共话成长与挑战
- ⑪ 推动中国开源发展的女性力量
- ⑫ “老奇葩”子寅: 不会“怼”产品经理, 干不了程序员
- ⑬ 程序员跨界有多猛? 他写了部130万字的科幻小说



扫一扫看视频
发现更多精彩



C++之父Bjarne Stroustrup: 程序员在数学上付出的努力永远不会白费

文 | Evrone 译 | 弯月

1979年诞生的C++已迈入“不惑之年”，却依旧在主流编程语言中占据重要席位。但今天，我们不再赘言C++的成功史，而将目光转向C++背后的英雄——C++之父Bjarne Stroustrup（以下称Bjarne）。在本篇文章中，让我们跟着软件开发商Evrone的采访，共享Bjarne Stroustrup四十余年程序人生中所蕴藏的大量经验财富。

作为最初只是用于改进C语言、增加部分面向对象编程功能的编程语言，C++的成功令Bjarne自身都感到意外：“C++的成功令人惊讶。”

不过，也是因为C++一直广受开发者喜爱，Bjarne多年来始终坚守在开发一线——目前71岁的他仍在积极参与C++的开发工作。经常有受益于C++的开发者说，C++改变了他们。殊不知，C++改变的还有Bjarne自己。

C++对C++之父的影响

Evrone: 你创建了一门高效且快速的编程语言，这无疑改变了我们的世界。对你个人而言，你在创建这门语言后，自身及工作是否也发生了变化？

Bjarne: 我从未想过这个问题，不过我认为这个问题可以从两个角度进行回答：哪些方面长期保持不变；哪些方面发生了巨大变化。

很久以前，我对历史、哲学等许多领域都抱有广泛兴趣，多年来保持不变，我认为这对C++的发展具有重要意义。因此，相较于深入研究学术，我更想去创建更多东西，这也使我对工程领域的看法带有一定“偏见”。我看重性能、低成本和可靠性，再加上我很重视反馈以及对现实世界问题的理解，最终造就了C++。

多年来，我在工作中持续稳步发展的一个方面就是教育。当我试图解释自己的想法时，我逐渐意识到：不能只是做出来，你必须教人们如何使用它。这也是C++一直存在的问题。通常，我的个人看法会被其他人的声音所淹没，这些人往往有着最简单的愿景却提出最大的主张。在20世纪八九十年代，经常



Bjarne Stroustrup 设计并开发了C++编程语言，美国计算机学会（ACM）及电气电子工程师学会（IEEE）的资深会员、美国国家工程院（NAE）院士。曾获英国工程技术学会颁发的法拉第奖章、美国国家工程院颁发的德雷珀奖（Charles Stark Draper Prize）等多个重要奖项。

会听到“我们没办法迅速培养教师”之类的抱怨，所以C++常常采用一种在我看来很糟糕的教学方式，也难怪有些人会对C++抱有非常负面的看法。

如果我能早点知道自己一生中大部分时间都在说和写英语，那么我一定会在英语上更加努力。这些年来，我需要面对并解决许多问题，这些问题种类繁多且范围广泛，在这过程中我发现许多困难的问题都与组织、管理、教育有关。与技术教育相比，广泛的阅读更令我受益匪浅。我明白了良好的语言设计需要高度谦逊，因为我们不了解的东西太多：这个世界在不断变化，我们的问题在不断变化，而我们自身也在不断变化。

解读行业当前现象及趋势

Evrone: 强大的C++元编程系统有时会被滥用，开发人员会想方设法将任务从运行时转移到编译时。在你看来，有没有什么可行的解决方法？

Bjarne: 每一种新颖且强大的功能或技术都会被过度使用，甚至错误使用，对此我想不到什么好办法。不过这也会带来一定好处，因为过度使用会让我们认识到问题所在，并设法解决。

举个例子，C++模板元编程非常有用，因此许多开发人员愿意忽略它的不足及负面消息去使用。后来，我们从总总结了大量经验，利用编译时求值的函数（`constexpr`和`constexpr`）和概念来弥补相关不足，大大简化了手动编写的大部分代码。

Evrone: 业界流传着一个“笑话”，即任何架构问题都可以通过引入新的抽象层来解决，结果却导致抽象层太多。我们看到很多C++代码都有大量的抽象类，那么作为C++创始人，你认为如何才能让每个抽象类都发挥积极作用？

Bjarne: 这就是David Wheeler的“计算第一定律”，他是我的论文导师，也是一个了不起的人，我从他身上学到了很多东西。

你说的这个“笑话”反映了现实，开发人员确实喜欢将真正的逻辑隐藏在多层接口之下，还会使用大量的间接方式，但这可能会导致代码量和运行时间增加一到两个数量级。因此，现代C++的大多数功能就是为了让人们编写的接口能被编译器优化成更简单的机器代码，尽量减少使用无谓的间接方式。

Evrone: DeepMind的AlphaCode神经网络问世后，业界声称这种神经网络很快就会取代开发人员。你如何看待？

Bjarne: 我不太确定。我也很怀疑人工智能是否真的能够取代开发人员，但我认为可靠性和最佳性能不太容易被标准化和平均化。AI不是我的强项，但TensorFlow和类似的库都有C++的身影，这也就意味着——不论好坏，起码我已经尽了自己的一份力。

Evrone: 在过去十年中，我们看到许多语法糖（对语言的编译结果和功能没有实际影响，却能更方便地使开发人员使用该语言的语法）被添加到主流语言中。通过“臃肿的语法”为开发人员提供更好的工具似乎已成为一种趋势，你如何看待这种趋势？

Bjarne: 只要能减轻开发人员的负担，即便是“臃肿的语法”也无伤大雅，我更愿称之为“让简单的任务变得更简单”。我认为，关键是让开发人员能直接在代码中表达基本思想。例如，用C语言风格的循环来表达容器上的简单循环没有任何优势或好处，最好使用Range-based for或其他算法。在大多数情况下，这些算法都能直接表达意图，只有在特殊情况下才需要考虑复杂的循环变量。例如，在访问容器的元素时，不是挨个访问，而是隔一个访问一个。想法越直接，编写、阅读和维护代码就越容易，优化也越简单。

因此，我认为“只有一种表达方式”不是理想的做法，这会导致有些情况很难被表达。此外，经过长年累月的优化，语言本身也会发生变化。从这两个角度来看，编程语言与自然语言并没有什么不同。

开发者如何摆脱困境

Evrone: 如今人们普遍认为，使用现代框架比应用数学知识更重要，你能在这方面给程序员一些建议吗？

Bjarne: 程序员在数学上付出的努力永远不会白费。学习数学是训练我们大脑的最佳方法之一，尤其是在与计算相结合时，我们很快就会意识到自己的错误。数学也教会了我们严谨，不要轻信过于简单或错误的想法。

在很多领域中，如科学计算、图形图像，以及许多金融软件中，数学知识必不可少。但对大多数人来说，数学的关键领域是概率和统计，很多人用数学来评价代码速度够快吗？能扩展吗？某些事件的发生概率是多少？影响是什么？

当然，也有许多应用程序不需要数学，但在构建基础设施或部署大规模应用程序时，容量和能耗的成本不容忽视，此时数学知识的不足就可能带来危害。

Evrone: 作为开发人员，有时面对手头的编程任务，我们总是列不出合适的解决方案。你有没有遇到过类似情况，可否分享一些处理这类问题的建议？

Bjarne: 当然遇到过。当面对新奇或重大难题时，每个人都需要付出巨大的努力，可能是几小时、几天甚至更长时间，我们可能还会陷入困境，感到无助和沮丧。

这个时候我们需要从逻辑层面来看问题。仔细想想你的目标是什么，或许是因为你努力的方向不正确，或许是因为你提出的要求不合理。也可以偶尔放松一下，想想别的事情。我一般会去跑步。放松下来后，脑海中就会浮现一些有用的想法。

Evrone: 业务环境时常要求我们在实现新功能时严格遵守最后期限。在你看来，开发人员如何在代码质量和开发速度之间保持平衡？

Bjarne: 这很大程度上依赖于管理结构和企业技术文化。我的个人建议（主要基于贝尔实验室的工作经历）是，如果你的组织内人才济济，不必要要求所有人都为下

一个截止日期而努力。你需要一些关键人物为未来做策划、开展实验、构建下一个重要系统的第一个版本及后续版本。我认为，一个好的组织需要构建稳定的产品线，团队中绝大部分人员都应在已部署并进入维护阶段的产品上进行升级。不过很显然，这种做法与常见的削减成本和发布颠覆性系统的思想背道而驰。

Evrone: 许多人将你视为自己的导师，你认为，一位优秀的导师应具备哪些品质？

Bjarne: 愿意倾听并认真理解问题。当提供建议时要谦虚，因为一般情况下，我们的理解都是不完整的。除此之外，优秀的导师给出的必须是具体建议，而不是一通模棱两可的废话。如果有人向你请教问题，你就应该给出认真的答案助其进步。但老实说，提供建议也是一件很难的事。

不过教学相长，这也是我们进步的主要动力，一位优秀的导师会从学生身上学到很多东西。

C++未来的发展

Evrone: 能透露一下C++未来的版本将会出现哪些变化？

Bjarne: 首先，必须适应C++20全新、强大和简化的特性，这一版本在C++11的基础上有了很大提升。C++20有很多语言特性，还有标准库组件等，在这里，我只介绍两个语言特性。

■ 模块。

我们认为，`import Mod`可以访问由`module Mod`导出（`exported`）的接口，这种方式更加整洁。与之相比，`#include`会泄漏实现细节和宏。此外，模块的编译速度也得到了大幅提升。例如，编译一段`import std`代码的速度比编译一段`#import <iostream>`代码的速度快十倍（注意：`std`包含所有标准库，而`<iostream>`包含的标准库不到10%）。`std`模块目前仍处于试验阶段，但经投票已被纳入C++23。

■ 概念。

在C++20之前，所有模板都是不受约束的。也就是说，它们没有定义供开发人员和开发工具查看模板的参数要求的接口，但C++20中提出的concept特性弥补了这一缺点。我们一直希望实现这种受约束的模板参数，只不过以前我不知道如何在不限灵活性或增加运行时开销的情况下实现这个想法。现在我们可以检查模板的使用情况，在出错时得到更加具体的错误信息，也可以重载函数模板，甚至及时提升某些方面的性能。

至于之后的版本，受疫情影响，很多功能的开发都被推迟了。我们希望能够C++23中推出一些重要的功能，但很可惜，我最喜欢的一些功能未能如愿进入C++23。在这里，我只透露三个功能。

■ 静态反射。

我们需要一种机制，可以根据程序中的类型在编译时生成代码，这不仅可以为我们提供运行时反射的灵活性，而且无须时间或空间成本。例如，我们可以针对一组固定类型，很容易地生成优化过的JSON查看器。在这方面，我们付出了巨大努力。

■ 模式匹配。

在许多函数式编程语言中，根据表达式与一组类型或备选值的匹配程度来选择操作，是实现模式匹配最方便的方法之一。我们可以在C++中实现相同的功能，且在此过程中无须使用switch语句。我们有一个非常完整的设计和实验性的实现，所以我非常期待C++26。

■ 并发模型。

多年来，我们一直在研究一个通用的并发模型，可惜我们发现的都是不太适合的用例，因此我们不得不推迟这一功能的到来。但我希望C++26能有所突破。

最后，有一点需要记住：单凭一个功能并不能让编程变得方便、安全且高效。我们需要在类型系统中创建大量特性。同时，我们又不能破坏现有的数十亿行C++代码，因为兼容性和稳定性非常重要。

本文已获原网站翻译授权，原文来自：

<https://evrone.com/bjarne-stroustrup-interview>

C#、TypeScript之父Anders Hejlsberg： 从事编程只因为热爱

文 | Darryl K. Taft 译 | 弯月

在诸多编程语言中，我们很难定论哪一门语言最好，不过其流行度或可从TIOBE榜单一探究竟。纵观该榜Top 10，大多诞生于20世纪，唯二较新的编程语言就是C#和TypeScript。这二者皆由同一人创建，他就是Anders Hejlsberg。在本篇文章中，我们将走进Anders Hejlsberg的程序人生，探寻其攀登至技术巅峰的真谛。



受访嘉宾：

Anders Hejlsberg

微软技术研究员，创建了包括Delphi、C#和TypeScript在内的多种流行编程语言，也是Turbo Pascal编译器的主要作者，还曾参与并领导了微软.NET Framework的开发。



采访者：

Darryl K. Taft

美国计算机协会（ACM）成员，拥有超过25年的业务经验，曾在各种科技出版社工作，包括政府计算机新闻（Government Computer News）、计算机系统新闻（CSN）、CRN、eWEEK和TechTarget。

鼎盛时期的摩城唱片公司不仅在节奏布鲁斯（R&B）上大放异彩，还制作了一系列热门歌曲，这些热门歌曲的背后是一众著名的歌手：史蒂维·旺德（Stevie Wonder）、史摩基·罗宾森（Smokey Robinson）、马文·盖伊（Marvin Gaye）、迈克尔·杰克逊（Michael Jackson）。

而如今微软在CEO萨蒂亚·纳德拉（Satya Nadella）的领导下，再次恢复活力并迈入全盛时期，这背后也有一位大名鼎鼎的人物：安德斯·海尔斯伯格（Anders Hejlsberg）。

Anders Hejlsberg，微软的技术研究员，也是一系列广受欢迎的编程语言的创建者，包括Turbo Pascal、Delphi、C#和TypeScript，其个人贡献影响了数百万程序员职业生涯发展。

编程界“帽子戏法”的上演者

微软开发部总裁Julia Liuson曾夸赞道：“Anders Hejlsberg是唯一一位创建了三种编程语言来完成‘帽子戏法’的语言设计师，他创建的每种语言都有数百万用户，如今我们喜爱的很多技术都离不开他。可以说，Anders Hejlsberg为提高开发人员的生产力做出了无私奉献，他为微软在开源领域所做出的贡献也无人能及。”

Madrona Venture Group董事总经理S. Somasegar（曾在Anders Hejlsberg任职期间担任微软开发部副总裁）也在邮件中表示：“Anders Hejlsberg是编程语言的大师级工匠，他的设计水平、开发经验，以及对编程语言性能的把握都无可匹敌。”

当前，TypeScript、C#和Delphi都是很受欢迎的编程语言，在Stack Overflow 2021年开发者调查中，TypeScript

和C#跻身最受欢迎的编程语言之列。当被问及最喜欢哪种语言时，Anders Hejlsberg表示：“这个问题太难回答了，就像问我最喜欢哪个孩子一样。”

提高开发人员的效率

Anders Hejlsberg表示：“我一直致力于提高开发人员的工作效率。考虑到程序员每天要在8小时的工作时间里使用编程工具，我认为他们对编程工具的热情无与伦比。遇到简单易用的工具，他们会爱不释手；对于复杂难用的工具，他们也会发表自己的看法。”

因此，Anders Hejlsberg对编程工具的开发充满了热情。在他看来，工具与开发人员的工作息息相关，工具是他们表达思想和算法的一种媒介，也是他们赖以生存的基础。

“这种爱与热情是让我坚持下去的动力。” Anders Hejlsberg说：“我选择从事编程，并不是因为想要一份工作，而是因为我对这个行业的热爱无可阻挡。我也不是为了赚钱才选择这份工作，相反为了这份工作我会不惜一切。”

合适的年龄，合适的时间

Anders Hejlsberg自认是一个很幸运的人，能够在合适的年龄以及在开发工具行业发展的初期加入科技行业，他在哥本哈根就读的高中也正好是丹麦第一批为学生提供计算机的高中之一——他第一次接触计算机大约是在1975年或1976年。

“当时还没有PC机，我们使用的是一台旧的惠普2100，带有32KB磁芯内存。这台计算机上有BASIC，还有Algol，这是我接触到的第一门编程语言。Algol语言是Pascal家族中的一员，是它将我引上了编程之路。”

后来，Anders Hejlsberg考上了丹麦理工大学（现在的丹麦科技大学），并在20世纪80年代初参与创立了一家

公司。该公司是哥本哈根最早的计算机经销商之一，人们可以在店内购买并组装属于自己的计算机。

在此过程中，Anders Hejlsberg发现人们在组装计算机时会遇到各种问题。因此，他们的业务模式开始转变为帮助客户搭建系统。为此，Anders Hejlsberg编写了很多软件，从而发现了自己的热情所在：“这让我最终走上了构建Pascal编译器的道路。也就是在那个时候，我加入了Borland（美国一家软件公司），后来的故事想必大家都知道了。”

在Borland工作期间，20岁出头的Anders Hejlsberg创建了Turbo Pascal，成为了该公司1983年的第一款产品。Turbo Pascal是一款Pascal语言的编译器，在IT专业人士和业余爱好者中广受欢迎。

Fullpower Technologies CEO、Borland联合创始人兼前CEO Philippe Kahn评价道：“Anders所做的一切都非常了不起。” Philippe Kahn曾与Anders Hejlsberg共事十多年，在他的印象中：“Anders Hejlsberg做事有条不紊，心思缜密，在架构和设计方面从不走捷径。他也非常全能，是很好的合作伙伴。在我看来，他身上有很多优点，我很怀念与他共事的美好时光。”

不仅如此，作为一名记者、作家、Windows Weekly播客的搭档主持、Thurrott.com网站负责人，Paul Thurrott也表示非常感谢Anders Hejlsberg和Turbo Pascal让他摆脱Amiga（Amiga公司开发的个人电脑品牌），转而使用DOS/Windows PC。

1995年，Anders Hejlsberg创建了Delphi编程语言和软件开发工具包。据了解，Delphi由Borland的Windows版Turbo Pascal演变而来，其开发环境是一个使用强类型高级编程语言的快速应用程序开发平台。

25年的微软从业经历

1996年底，在Borland和微软这两家公司发展的关键时刻，Anders Hejlsberg却选择离开Borland，加入微

软（2021年11月，微软还为他庆祝了加入微软25周年纪念日）。

当时，Borland是微软在软件开发工具领域强劲的竞争对手，时任微软联合创始人兼CEO的Bill Gates（比尔·盖茨）非常重视开发人员，由此发起了一场招聘热潮，最终导致Borland于1997年因人才流失问题向微软提起了诉讼。直到1999年，两家公司才平息了这场风波，私下达成和解：微软向Borland投资2500万美元，两家公司达成1亿美元的合约，由Borland为微软的核心技术提供授权。

2000年，为了与Java竞争，Anders Hejlsberg在微软创建了C#语言。C#是一种通用的多范式编程语言，按照维基百科上的介绍，C#包括静态类型、强类型、词法作用域、命令式、声明式、函数式、泛型，以及面向对象、面向组件等编程原则。

与此同时，2002年微软还发布了开发框架.NET Framework（最初是专有技术，但目前已开源）。.NET Framework是通用语言基础设施的主导者，直到被跨平台.NET所取代，而微软推出.NET也是为了与Java竞争。

后来，在Anders Hejlsberg的领导下，微软于2012年推出了TypeScript。TypeScript是JavaScript的语法超集，为其带来了可选的静态类型，专为开发大型应用程序设计，可转译成JavaScript。

TypeScript作为Anders Hejlsberg最新创建的编程语言，建立在流行的JavaScript基础上，因此TypeScript也可能是Anders Hejlsberg所创建的语言中最受欢迎的一个。Anders Hejlsberg对此表示：“这些项目、Typescript的开源活动、我在这个过程中的收获，以及开源社区开发人员的热爱，都是让我坚持下去的原因。”

对于以上经历，Anders Hejlsberg自己也感慨道：“非常精彩，这是实现自己梦想的方式：打造一款大获成功的产品。这期间微软给予了我很多帮助。”而Borland前CEO Philippe Kahn对此也没什么不满，只是开玩笑说：

“我觉得Anders Hejlsberg在微软工作的时间太长了，

他应该更像一个企业家。”

如今，Anders Hejlsberg仍在深入参与TypeScript开源项目，每天都在积极编写代码——精力主要集中在TypeScript工具的核心部分“类型检查器”上。此外，Anders Hejlsberg也在积极参与C#的发展及其语言设计团队的工作：“我会经常与微软内部，以及外部从事编程语言和编程工具相关工作的其他团队交流。”

最喜欢的编程语言

Anders Hejlsberg表示：没有最喜欢的编程语言，但现在主要在用JavaScript编写代码。

Anders Hejlsberg解释道：“我无法说哪个才是我最喜欢的编程语言，所有语言我都挺喜欢的。刚开始，我几乎和所有人一样，都是先接触BASIC，后来用Algol编程，还学习了Pascal。之后又用Pascal编写代码，还开发了一个Pascal编译器。但其实早期我用汇编语言写了大量代码，因为那时只有使用汇编语言才能获得最佳性能——8位计算机的资源非常有限，只有64KB内存，为了将所有功能都压缩到极少的内存中，你不得不编写机器代码，只有这样才能编写出简洁又紧凑的代码。当然，这种方式的效率很低，调试也非常不容易。”

所幸，如今计算机的速度提升了几万倍，存储空间也增加了几十万倍。

“这些巨大的提升简直不可思议，某种意义上，如今计算机拥有取之不尽的资源，不再需要手动优化机器代码并将其放入少之又少的内存中。”Anders Hejlsberg表示：“如今，我们想方设法，尽可能让机器完成大量工作，让程序员的精力集中到更为重要的创新部分，这也就是为什么程序员的工具会变得如此重要的原因。例如，想重命名某些函数，手动搜索和替换不仅速度慢，错误率也很高。程序员应该利用工具进行重构，遵循类型检查器的建议。”

Anders Hejlsberg进一步补充道：“现在我们有自动补齐

工具，在输入语句时会自动显示列表供你挑选，程序员无须再翻看手册去查找某个对象可以调用的方法名称。此外，这些工具还拥有展示参数及其他实用功能。在这些工具的帮助下，程序员可以集中精力思考创新、算法及其他机器无法完成的工作，即更能体现人类价值的一系列工作。”

虽然Anders Hejlsberg没有最喜欢的编程语言，但他坦言Visual Studio Code是他最喜欢的工具：“我每天都在Visual Studio中编写TypeScript代码。有意思的是，TypeScript语言本身就是用TypeScript编写的，Visual Studio Code也是用TypeScript编写的。因此可以说，TypeScript完全是自力更生，它是一款很棒的工具。”

TypeScript盛行

在Vercel CEO Guillermo Rauch看来，TypeScript已成为当前JavaScript技术栈的首选工具。在TypeScript的助力下，JavaScript生态系统和Next.js终于从小型项目发展成为可与Java或.NET生态系统并驾齐驱的企业级平台。

Guillermo Rauch指出：“TypeScript支持增量式的采用过程，因此所有JavaScript开发人员都可以轻松使用TypeScript。如今我们可以看到越来越多的开发人员在创建新项目的时候，都选择了TypeScript。Anders Hejlsberg所开发的TypeScript帮助解决了JavaScript的许多问题，这不仅对前端软件开发产生了深远的影响，也改变了一代开发者的命运。”

除此之外，Philippe Kahn的Fullpower-AI项目（致力于为医疗和健康行业客户提供先进的物联网生物传感解决方案）中，存在一些用JavaScript组件的遗留解决方案，对此他表示：“我们选择使用TypeScript进行开发，因为它的开发环境更好，还可以编译成JavaScript。Anders Hejlsberg开发的TypeScript堪称一流。”

开发工作流程的演变

与职业生涯初期构建Turbo Pascal的工作流程相比，

Anders Hejlsberg直言他现在最常用也更喜欢的是GitHub的工作流程。

“构建Turbo Pascal时，我在哥本哈根，但部分团队成员在美国的斯科茨谷（加利福尼亚州，Borland总部）。当时，能买到的最好的电话调制解调器是1200波特，基本上是每秒120个字节，这在当时已经非常快了”，Anders Hejlsberg回想：“即使是开发Turbo Pascal这样的‘小型’产品，也无法展开团队合作，因为你根本无法传输这些资源。当时最快的方式就是通过快递，在丹麦和美国之间传递装有最新版本编译器的软盘。”

对比如今，他十分感慨，“回头再看看如今的工作流程，即便受疫情影响，我们的团队成员也没有错过任何信息，因为一切都在云端。我们开源项目的整个工作流程都在GitHub上，召开会议则可以使用Teams、Zoom或Discord。”

拯救程序员的职业生涯

Anders Hejlsberg说，经常有人告诉他“你的工作拯救了我的家人，也拯救了我们的生命”。

举个例子，有人曾对他说，苏联的突然解体导致许多人都失业了，但后来Turbo Pascal、Delphi的兴起让他们重新找到了工作。他们可以用Turbo Pascal或Delphi为当时刚刚起步的创业公司编写程序等。

“他们认为我是他们的救命恩人，这让我意识到自己的工作是如此重要。”Anders Hejlsberg认为：“为这个世界贡献自己的力量，是每个人的理想，不是吗？起码要在我们离开这个世界之前，给世界带来一点改变。”

他人眼中的他

毫无疑问，Anders Hejlsberg对这个世界的贡献良多，他不仅影响了开发人员和微软的客户，对团队成员和曾受教于他的其他人也有很大帮助。

Miguel de Icaza是一名杰出的微软工程师，也是GNOME、Mono和Xamarin开源项目的创始人。他与Anders Hejlsberg合作的次数其实并不多，因为目前Anders Hejlsberg主要负责TypeScript，而他本人主要参与C#项目。

尽管如此，Miguel de Icaza仍表示：“Anders Hejlsberg给我留下了深刻的印象。他是一位出色的导师，他的解释清晰且富有启发性。当你提出一些语言设计方面的想法时，他会帮你迅速分析哪些想法不太理想、哪些想法很好，并给出相应理由。”Miguel de Icaza还在邮件中写道：“Anders Hejlsberg对JavaScript和Web也充满热情，他非常喜欢其交互模型，灵活性，以及借助这些技术赋予人们更强的能力。”

Pulumi CTO Luke Hoban也曾与Anders Hejlsberg在微软共事过一段时间（2003年Luke Hoban作为实习生加入微软，直到2008年他都一直与Anders Hejlsberg一起从事C#的工作，2010—2013年又转战TypeScript的开发工作），他说：“在我的整个职业生涯中，Anders Hejlsberg一直是我的榜样。他的直觉、思考方式和设计美学在编程语言和开发工具领域确实是独一无二的。他是一位了不起的同事，能够为项目增添活力和乐趣。”如他所说，Anders Hejlsberg本人非常风趣，比如在Visual Studio 2005的一次活动上，他就装扮成了喜剧片《王牌大贱谍》中的主角特工奥斯丁·鲍尔斯。

Luke Hoban表示：“在与Anders Hejlsberg共事期间，我发现每次与他在设计问题上意见相左，几乎无一例外都是我自己的问题，只是当时我还没想明白。”

为开发人员的切身利益考虑

Luke Hoban指出，在过去25年创建的编程语言中，只有C#和TypeScript这两种至今仍位于十大最常用语言之列。而这两种语言的首席设计师都是Anders Hejlsberg，这实在令人敬佩。

“C#和TypeScript之间有很大的不同，但二者有一个

共同点：开发者整体的体验都很棒，不仅是语言本身，还有与语言相关的工具。”Luke Hoban回忆道：“我记得，在为C# 3.0设计LINQ期间，C#添加了类似SQL的语法，Anders Hejlsberg坚持认为FROM、SELECT子句的顺序应该与传统的SQL顺序相反，这样才能让开发人员感受到流畅的IDE使用体验，因为这个顺序更符合用户输入这些表达式的习惯。”

Madrona Venture的Soma Somasegar也曾多次目睹这类场景。在面对两种不同技术、团队，或两种不同技术方法而难以抉择时，Anders Hejlsberg总是能够站在正确的一方，为客户（即开发人员）的切身利益考虑，并提出优化方案。“即使正确的选择是不太受欢迎的团队或方法，他也毫不动摇，因为这一切都是为了替客户着想，做正确的事情。”

为此，Soma Somasegar将Anders Hejlsberg称为“编程语言之父”，并称“坚持为开发人员的切身利益考虑，这种热情贯穿了他的思维过程、设计和开发工作以及他所做的一切。我记得有几次，在快到截止日期时，他仍坚持推出某个功能或修复某个漏洞，就因为这对开发人员来说是正确的事情。我非常欣赏和钦佩他的这种精神。”

推动开源发展

某种程度上来说，Anders Hejlsberg是推动微软追求更开放的开发流程、构建开源项目和为开源项目做贡献的核心人物。微软TypeScript高级项目经理Daniel Rosenwasser表示：“在推动微软开发部门朝着开放式开发迈进方面，Anders Hejlsberg发挥了重要作用。在他的带领下，C#和TypeScript才取得了成功。”

出于对开源的热情，Anders Hejlsberg甚至表示：“老实说，如果微软没有做出这种转变，我甚至会怀疑微软是否还能发展下去，我个人肯定也不会继续留在微软了。但如今我们都在认真地努力，争取发展得越来越好。”

在Anders Hejlsberg看来，开源与微软传统开发方式之

间的最大区别在于，开源让微软无限接近用户。“你不必再秘密制定产品计划，也不必秘密举行设计会议。你可以公开所有工作，让每个人都参与进来。这种方式不仅创造了一个更好的工作流程，也提升了为客户、用户提供满意服务的能力，因为每个人都可以针对议题进行投票。”

改变微软的业务发展

Blue Badge Insights CEO Andrew Brust认为，在Anders Hejlsberg来到微软之后，随着C#和.NET Framework的推出，微软在编程语言与平台方面的整体水平都得到了很大提高，公司的整个开发栈也发生了改变。此外，Anders Hejlsberg后来做出的贡献，如Roslyn (.NET编译器平台) 和TypeScript在某些方面更是有着惊人的表现，这表明Anders Hejlsberg不是一个故步自封的人。

不过，Andrew Brust觉得这也有一定缺憾：“在我看来，在.NET出现以前，微软通过Visual Basic构建了庞大而又忠实的快速应用软件开发生态，其主要精力都集中在服务这些客户。后来由于Anders Hejlsberg对工业级编程的敏锐观察，使微软的关注重心发生了改变。然而，昔日的快速应用软件开发就相当于如今的低代码和无代码，可微软直到最近才通过PowerApps重新在此领域站稳脚跟，并仍在加紧追赶。造成这种局面的原因，除了Anders Hejlsberg的光环外，还有微软长期以来很难同时推进多条战线的历史性问题。”

Ted Neward是Neward&Associates的负责人，长期兼任.NET和Java开发人员和培训师，在他看来，Anders Hejlsberg的职业生涯发展可以分为三个阶段：Delphi、C#和TypeScript。

“每个阶段，他都会做一些与主流、面向对象或强类型有所不同的尝试，而每次的环境和目标也都不同。Delphi的重点是二进制组件 (VCL)；C#解决了传统的面向对象问题，同时还推出了更多特性和功能；而TypeScript构建在无类型/动态类型的运行时之上（与前

两个不同）。”

Ted Neward表示，有些人可能会认为C#包含的功能过多。但总体来看，Anders Hejlsberg从未有太过偏离主流的思维，他借鉴了其他范式的思想，在创建编程语言的同时还考虑到了趣味性、热门话题和新鲜感等方面。

不过，Ted Neward认为Anders Hejlsberg“可能永远无法像Guido van Rossum (Python之父) 和Larry Wall (Perl语言之父) 一样，广受粉丝的追捧。”但不可否认的是，Anders Hejlsberg是迄今为止最成功的语言设计师，也是世界上绝无仅有、一生创建了三门主流编程语言的人。

依旧谦逊、勤奋

Anders Hejlsberg十分热爱自己的事业，尽管已成为业内的佼佼者，但他待人处世依然谦逊有礼。

在访谈最后，Anders Hejlsberg总结道：“我非常擅长编程，我喜欢复杂的东西，也喜欢没有太多依赖的东西。编译器和编程语言有点相似，因为它们都是底层的技术。从某种意义上来说，只有创建出一门编程语言并开发工具，你才能成为自己编程世界的主人。”

本文已获原网站翻译授权，原文来自：

<https://thenewstack.io/anders-hejlsberg-microsofts-app-dev-star-maker-machinery/>

MySQL之父Monty：代码是一次写成的

文 | 郭露

MySQL之父Michael "Monty" Widenius（通常称为Monty），是开源数据库运动的倡导者之一，曾一手创办MySQL和MariaDB。有着四十多年编程经验的他，如今依然每天坚持写程序，并且总结出了一套对技术的独到看法。

如今，我们正处于被数据重新定义的时代，软件、企业的信息化，万物互联等浪潮背后都离不开数据的支撑。而随着开源理念的深化，越来越多的数据库选择创建开源社区，让更多开发者参与到数据库的建设中来。MariaDB就是其中较为重要的开源数据库之一，而它的创建者正是大名鼎鼎的Monty。

Monty (Michael "Monty" Widenius) 是开源软件运动的著名倡导者，也是MySQL（他写下了MySQL的第一行代码）和MariaDB的作者、MySQL AB公司的首席技术官，以一人之力完成了MySQL数据库服务器中95%的代码。作为在技术界历练半生的资深程序员，

Monty一直与代码相伴，时至今日，年逾60岁的他仍然每天坚持写代码。正因如此，他对编程有着许多独到的见解。在他看来，只有沉浸编程超过20年，才能“像欣赏音乐一样，领略编程之美”。另外，他还认为“代码要尽量一次写成，避免多次修改”。只有做到这一点，才能称得上是一名优秀的程序员。



Monty MySQL之父

“我在编程方面有一定的天赋”

1962年，Monty出生在芬兰首都赫尔辛基，那时，计算机的发展还处于萌芽阶段，互联网则是以“内网”的形态用于军事或学术领域。1971年，Kenbak-1面世，标志着个人计算机的崛起。接下来的十多年里，计算机不断迭代，市面上出现了HP 3000、IBM 5100及Apple I（见图1）等热门产品。随着计算机的体积越来越小，它才逐渐走进大众视野。

尽管如此，当时的计算机依旧属于“高档货”，但16岁的Monty在见识到计算机后，下决心要自己攒钱买一台。于是他利用暑假在马路上铺沥青做兼职，最终在父亲的资助和自己的努力下拥有了人生中的第一台计算机——ABC80 (Atanasoff-Berry Computer, 即阿塔纳索夫-贝瑞计算机，通常简称“ABC计算机”)。尽管ABC80 (见图2) 的CPU只有4MHz，内存也只有8KB，但对Monty来说，这已经十分奢侈了。正是在这台计算机上，Monty用BASIC语言写下了第一行代码，从此与编程结下了不解之缘。

在学习计算机的初期，Monty



图1 Kenbak-1、HP 3000、IBM 5100及Apple I (图源自Wiki)



图2 ABC80: 瑞典公司 Luxor 计算机系列中的第一台计算机 (图源自Wiki)

投入了大量的精力。当时ABC80上的软件很少，Monty便通过汇编语言试着让游戏能够在上面运行，其中就包括1978年推出的经典游戏《太空入侵者》。这是他第一次将自己所学的编程知识运用到实际问题中。以至于多年后回忆起自己的这段学习历程，他仍然直言“自己在编程方面有点天赋”。

随着技术的深入，ABC80的内存空间和运行速度开始无法满足Monty的需求。于是两年后，他便将ABC80升级为ABC800，并开始开发真正的功能性软件，包括文字处理器、硬盘控制器及磁带备份软件等。

由于ABC计算机为瑞典制造，因此Monty在开发的过程中经常需要去瑞典买配件。某天他偶然走进了Allan Larsson (以下简称Allan) 的计算机商店，作为那个时

代少数能开计算机商店的人，Allan也是个编程爱好者。Allan给了Monty一个数据库程序，想让他帮忙为客户开发一个软件。而Monty认为Allan的这个程序太复杂了，并且不实用，于是他直接重新写了一个数据库，即后来的Unireg。

经过这件事之后，Allan对Monty的能力非常认可，这为他们之后的合作奠定了基础。1985年，Monty和Allan成立了TCX DataKonsult AB公司，这也是最早的数据库公司之一。

到了20世纪90年代，互联网开始飞速发展，Web技术应运而生。不少客户要求开发基于Unireg的Web应用，但一方面，当时Unireg属于桌面软件，不支持SQL，也不能和Web进行集成；另一方面，当时的数据库系统已逐渐发展成熟，市面上出现了IMS、System及PostgreSQL等数据库。因此，Monty认为公司应当顺应潮流。于是在1994年，他将SQL接口添加到Unireg中，并将其更名为MySQL。1996年10月，MySQL正式版本发布，从此掀开了数据库历史的重要篇章。

MariaDB的诞生

随着Web的兴起，MySQL也针对Web推出了多个版本。团队规模也从最初的3人变成了15人。1999年，MySQL AB成立，其发展吸引了资本市场的注意，收购价一度高达5000万美元（被Monty拒绝）。到了2003年，MySQL实现了400万的安装量，较两年前翻了一番，成为当时全球最受欢迎的开源数据库。

2008年，MySQL的年收入已经高达7500万美元，为了获得更好的发展，公司决定接受Sun Microsystem的收购。一来Sun Microsystem是一家技术型公司；二来它也是开源运动的推动者之一，被它收购可以让MySQL保持开源。但没想到的是，在被Sun Microsystem收购一年后，Sun Microsystem就以74亿美元被Oracle收购了。

担心MySQL可能会受到Oracle的控制而变得商业化，Monty发起了一场拯救MySQL的请愿活动，并直言不讳

地说：“作为MySQL的核心开发者，我们不相信Oracle能够善待MySQL，我们要确保它是永远开源的。”最终在MySQL闭源之前，Monty将其分化，创建了MariaDB（见图3），以自己小女儿Maria（见图4）的名字命名。

Why MariaDB was created

“Save the People, Save the Product”

- To keep the MySQL talent together
- To ensure that a free version of MySQL always exists
- To get one community developed and maintained branch
- Work with other MySQL forks/branches to share knowhow and code

After Oracle announced it wanting to buy Sun & MySQL this got to be even more important.



图3 设计MariaDB的初衷：维护大众权益，守护数据库产品（图源自MariaDB官网）

Maria then and now



图4 Monty的小女儿Maria（图源自MariaDB官网）

MariaDB开源数据库可以看作MySQL的一个分支，主要由开源社区维护，目的是要完全兼容MySQL，包括API和命令行。在MariaDB推出后，不少MySQL的老员工选择追随Monty加入MariaDB。甚至原先使用MySQL的公司也将数据库迁移到了MariaDB上，其中就包括谷歌和维基百科。对此Monty解释道：“与MySQL相比，MariaDB更加成熟，拥有更大的研发优势，并且在安全性修复方面也更加出色。”直到现在，Monty依旧亲自参与MariaDB的开发维护，将自己的工作重心都放在了MariaDB上。

MariaDB坚持开源的背后

《新程序员》：你在创建MariaDB时，曾提到要把它打

造为第二个MySQL，并且要确保它是开源的。对于数据库而言，为什么开源这么重要呢？

Monty: 对于任何大型项目来说，开源都是非常重要的。既然要和巨头竞争，就要有和他们一样的工具。在我看来，开源有利于软件开发，尤其是当公司规模还不小时。这个时候你很难兼顾公司和用户的需求，因此需要听取别人的想法。而开源就意味着可以获得社区的帮助，能够了解其他人的观点。开源后，你可以开发出更好的产品，同时产品也能够产生更大的影响力。

《新程序员》：开源有个弊端就是声音太多，需求不一，这种情况下该如何保证数据库能满足大多数人的需求？

Monty: 要解决这个问题，就需要确保数据库足够灵活，这样才能满足大多数人的需求。在这一点上，MySQL和MariaDB的做法是建立各种性能不一的存储引擎，人们可以针对具体需求开发自己的存储引擎。

事实上，对于那些有需求的人来说，MariaDB依旧是一个优秀的工具。而对于要求数据库体量较小且运行速度较快的人来说，MariaDB同样是一个不错的选择。在开发MariaDB时，我们考虑到了各种可能性，使它能够保持良好的性能。

《新程序员》：AI技术的发展让人们们对数据库的期待发生了转变，今天数据库是否能与AI技术结合，从而拥有数据决策能力？

Monty: 对于数据库来说，最重要的是处理AI需要的数据结构。因此我们添加了对JSON的支持，使MariaDB支持动态列。这样人们就可以存储并检索数据，同时保留自己想要的格式。通常AI并不是要创造内容，而是要实现文件自动化，这也是我们对于MariaDB所抱的期望。因此数据库和AI完全是不同的工具集。

除此之外，我们还需要一个良好的环境，其中每一部分都是可替代的，要确保自己不被束缚。一旦有了束缚，你的应用程序就需要与静态系统相结合，这会大大降低

灵活性。我认为对于数据库来说，要注意的一点就是，要确保数据库容易上手，而这恰恰意味着更多的AI技术能够整合到数据库中。

每天坚持写代码

《新程序员》：在中国的IT行业有这样一种现象，认为程序员过了35岁就要转型，进入管理层或其他领域。对此你怎么看？

Monty：这在很多地方都很常见。这个现象的主要原因是程序员在管理岗位上的工资要比单纯做编程高。很少有公司会重视优秀的程序员，这就导致了收入的差异。我认为，如果程序员没有晋升的空间，与其让他们被迫转型，不如建立一个能提升他们收入的新环境。要想做到这一点，公司就得让他们承担更多的责任。要程序员担任管理岗位也行，但前提是仍然要保证他们每天写代码的时间。毕竟好的经理人到处都是，好的程序员却千里挑一。

《新程序员》：据我所知，你每天仍然在坚持写代码，但同时也要负责MariaDB的运营和管理。你如何平衡这两种工作呢？

Monty：在写代码这方面，我还是有一点天分的，所以我想坚持下去。我会雇佣经理人为我工作，这样我就可以做我最擅长的事情。我会参与代码审查、社区运营及MariaDB的相关决策。但同时我也会花很多时间维护客户，与不同国家的开发者交流，其中也包括许多中国的开发者。我认为，除了写代码，这是我做的最重要的事。

《新程序员》：听闻你很早就开始在家办公，如今这一办公方式也开始流行起来，对于远程办公你有什么看法？

Monty：事实上我认为远程办公是非常灵活的工作方式，从1981年开始我就在家办公（MySQL和MariaDB团队都是在家办公）。我们在招聘之前可能从来没见过候

选人，甚至都不知道对面是个人还是团队。但是我们的效率一直在线。做到这一点的前提是，要对与自己联系密切的同事足够了解，至少熟悉他们的样貌。

我认为对80%的开发者而言，在家办公是个不错的选择。可能有一小部分开发者，他们的工作负担比较重，在家提不起精神来。这时就需要他们出去走走，见见朋友，或接触接触新事物。我刚开始在家办公时，也会担心会不会被孤立的问题。所以后来我会定期在家里举行派对，也会亲自下厨。团队每年也会在一起待一段时间。

一个非常出色的程序员能顶五个一般的程序员

《新程序员》：对于你来说，过去几年数据库领域发生了哪些大的变化？

Monty：在过去的5年间，学习SQL（结构化查询语言）成为一种趋势。但是人们发现SQL过于复杂，还需要学习其他更容易上手的语言来解决问题。于是许多公司开始创新，采用NoSQL（非关系型数据库）进行开发。但是，人们逐渐意识到NoSQL并不是“万金油”。那么，是否要选择关系型数据库？关系型数据库是否能涵盖NoSQL提供的功能？很明显，有的可以，有的不行。因此我认为，当下环境中，对数据库的要求是要保证云端及本地部署。

不要被一个数据库束缚。云端提供的是灵活性，能在数据库中运行软件，即使是成百上千个软件，而本地部署的价格更低，控制权限更高，这一点是云端无法提供的。但我依然认为云端有它的优势，我们要在两者之间找到平衡。

《新程序员》：30年前，人们说到数据库，一般是指去银行办理业务。现在，人们有了更多的选择，可以借助数据库实现许多功能。而谈到数据库开发，人们往往指的是“后端开发”。对于想要进入数据库领域的开发者或

毕业生，你会给他们怎样的职业建议？

Monty: 在我看来，选择开源数据库来入门更简单。现在开源数据库很多，如果你想达到专家级别，或得到一份好工作，可以选择一个合适的数据库，并学习如何进行优化。同时，你需要了解人们的需求，可以和从事这一行的同学交流，学习解决数据库中的实际问题。

《新程序员》：除了多参与开源项目之外，对于中国开发者你还有哪些想说的？

Monty: 我和来自中国的开发者有过非常多的互动，他们非常出色，在编程上表现得非常优秀。不过我在感到惊喜的同时，也感到非常惋惜，因为他们都想转型做管理。我认为这是最大的错误。他们需要让老板给自己派更多的任务，当然也可以做管理，但前提是能让自己写代码。还是那句话：找到一个好经理很容易，但找到一个好的程序员很难。一个非常出色的程序员可以抵五个一般的程序员，关键是你想当一个好的程序员还是一个平庸的经理。对于所有中国开发者，我只想说，请坚持你的工作，你已经做得非常好了，一定不要停止写代码。

参考资料

- [1] <https://zh.wikipedia.org/wiki/%E7%B1%B3%E5%8D%A1%E5%9F%83%E7%88%BE%C2%B7%E7%B6%AD%E5%BE%B7%E7%B4%90%E6%96%AF>
- [2] <https://blog.openocean.vc/founder-stories-a-hackers-hacker-6d5054c90564>
- [3] <https://huskyintelligence.com/leverage-open-source-code/>
- [4] <http://monty-says.blogspot.com/2009/12/help-saving-mysql.html>
- [5] <https://www.geeksforgeeks.org/introduction-of-mariadb/>
- [6] <http://www.josetteorama.com/from-mysql-to-mariadb-michael-%e2%80%9cmonty%e2%80%9d-widenius-talks-about-databases-and-his-projects/>
- [7] <https://dri.es/the-history-of-mysql-ab>
- [8] <https://mariadb.org/wp-content/uploads/2019/11/MySQL-MariaDB-story.pdf>



扫码观看视频
听Monty分享精彩观点

责任编辑: 杨静华

PROGRAMMER 新程序员

.004

成就一亿技术人



中国水利水电出版社



人人都是程序猿
“专属微信服务号”

ISBN 978-7-5226-0689-7



9 787522 606897 >

定价: 89.00元

销售分类: 计算机科学