

# 高端的编程语言是否真的高端？

作者: [ztuio](#)

原文链接: <https://ld246.com/article/1501202128369>

来源网站: [链滴](#)

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

我看到了许多刚接触Ruby、Python和Javascript的新手程序员抱怨这些语言太低端，就好像是在以某种方式欺骗系统一样。他们认为他们没有学习到那些真正“高端”的编程语言。某些语言是否真比另一些语言更高端？我们一起来研究一下。

## 从电频层面上看

众多的编程语言在方方面面都有所不同。

但从低级的机器语言到我们大多数人使用的高级语言，都离不开电频这项技术。低使用性的机器语言没有表现得比那些高级代码更高端或者先进。低使用性是众多语言设计者在完成特定设计目标的长期择结果。

让我们继续深入，这里有一个用机器语言编程的例子：

这段代码一旦被数字处理器解析为电子信号后就能直接操作数字传感器和电动机了。在#之后的那些可读懂的东西会被电脑忽略——它们之所以被程序员放在这里是为了让其他人读到这段代码后能知道究竟在干什么。感谢上帝，我们不用必须像这样来编写代码。

程序要控制电脑的话最后都会转变成机器语言的。但是我们创造了编程语言用来取代那些直接用二进制、十六进制或者十进制代码来编写的机器语言（如早期的计算机科学家像阿兰·图灵和他朋友所做的那样），它帮助我们以更快更容易的方式将自己的想法转化成机器指令。有许许多多多种方式都可以生成你上面所看到的那段机器语言，因此就会有许许多多多种编程语言，但是它们全部都会在某处转化为电频谱。

从这个角度来看，语言的选择其实就是对让电脑帮助你生成机器语言的方式的选择。那么，高端和低端，区分的依据何在？

## 从设计角度上看

今天所有的程序语言都是人们为了完成某种用途而设计出来的。这种理解是至关重要的，因为它们的计目标决定了当程序通过它们运行时用户将得到的体验。

让我们通过设计目标来对比两种编程语言C和Ruby。

- C的目标：接近于机器语言，但能够移植在许多机器架构上。
- Ruby的目标：美丽、简单以及令开发充满快乐。

虽然最终都会生成机器语言，但不同的设计目标使这些语言的使用体验彻底不同。由于C语言结构更近于机器的实际工作，因此它的机器代码在计算机执行时会更加高效、快速。这里有一个记录了各种可视化程序语言在处理某个简单文本的速度的图表，随着文本尺寸大小的变化，程序语言在处理速度上生了变化：（纵轴为处理花费的时间，横轴为文本的大小）

由于执行时间很容易衡量（人们喜欢优化他们能够衡量的东西），象C一样“快速”的语言在人们精层面上就会认为是高端的编程语言。

这其实是虚假的，这与不精确的速度测量密切相关。

## 速度的测量

C和那些底层语言在计算机硬件迟缓以及系统资源受限的时代是不可或缺的。毁灭战士、雷神之锤以其它的第一代3D游戏的出现才突破了只能使用注重机器运行速度的语言的限制。

然而在今天，相对于我们在计算机上需要的大部分东西，计算能力是廉价和充足的。在软件开发中真正受限的是开发人员的时间和精力。象Ruby那样的语言可以让程序员用更少的代码做更多的事情，使发更加快速。增加开发速度意味着可以更快的找出你所做的是否是真正的需求，可以更加简单的添加户要求的功能以及可以使用更少的开发人员更轻松地完成项目。

## 多元化

这是一个错误的命题，因为问题的答案取决于你的需要。科学家、统计学家和数据分析师仍然需要一象C语言那样注重机器运行速度的语言来帮助处理他们的工作。大多数网络开发者需要象Ruby或者Python那样注重开发速度的语言来快速满足他们的业务和客户目标。在每个情景中，“最好”的语言都是同的。

需求的多样性支撑了一个在不同设计目标下，程序语言多元化的重要性的强力论证。程序设计中永远会有一门人人都采用的世界语，甚至是机器语言也非潜在的通用语言，因为每台电脑的架构都采用了有不同的机器语言。对于对软件有不同需求的我们来说，这是一件好事。

## 独立的认识

在新手程序员阅读到这篇文章的时候：我希望它可以帮助你对所学习和所书写的语言代码有自己独立认识。对于C语言，你需要知道它更贴近于机器语言而不需要了解它是看起来更“高端”的语言。我创造编程语言的目的是为了减轻自己的负担，能够更加快速方便的创造想要的东西。

如果你要选择学习C语言，你选择的目的是应该是它擅长对Arduino和嵌入式单片机编程或者是它能够速编译机器算法，而不是你认为它比你已经学过的语言更加“高端”。让你想从事的项目来决定你要择学习的语言。因为毕竟它们都只是机器语言中的其中一种方言。

## 合适的选择

我所认识的最高端的那些程序员从从事编程工作开始就爱做一些很浅显的事（他们中的很多人如今仍保持这种习惯）。他们会自己做一些很简单的游戏，或者做一些搜索和可视化的实验。他们所做的每事情都来自于他们对此的兴趣，编码对他们来说就像是在玩耍一般，这非常的有趣。

不断重复对相关问题的反馈是掌握任何技能的关键。无论你选择的语言或项目是什么，请确保花费大时间来学习它并听从指导者的建议反馈。代码语言产生的真正目的是为了帮助别人建立或者完善那些棒的东西。无论什么语言，你的选择应永远是能够为你提供最快捷、最有效帮助的那一种。