

# 《代码英雄》第二季（2）： Hello, World

代码英雄讲述了开发人员、程序员、黑客、极客和开源反叛者如何彻底改变技术前景的真实史诗。

## 什么是《代码英雄》

Command Line Heroes

代码英雄是世界领先的企业开源软件解决方案供应商红帽（Red Hat）精心制作的原创音频播客，讲述开发人员、程序员、黑客、极客和开源反叛者如何彻底改变技术前景的真实史诗。该音频博客邀请到了谷歌、NASA 等重量级企业的众多技术大牛共同讲述开源、操作系统、容器、DevOps、混合云等发展过程中的动人故事。

本文是《[代码英雄](#)》系列播客[第二季（2）：Hello, World](#)的[音频](#)脚本。

导语：每一种新的编程语言的诞生，都是为了做一些以前无法完成的事情。如今，有非常多编程语言可以选择，但哪些才是你真正需要了解的？

本集将深入探讨编程语言的历史。我们将会了解到被称为“神奇葛丽丝”的天才——海军少将葛丽丝·哈伯。多亏了她，开发者不再需要数学博士的学历就能使用机器代码编写程序。参与录制的还有 Jupyter 项目的 Carol Willing，她是 Python 基金会的前理

事；以及 New York Times Magazine 和 Wired 的撰稿人 Clive Thompson，他最近正在撰写一本关于程序员如何思考的书。

**00:00:07 - 各种语言：**

“你好，世界。”

**00:00:12 - Saron Yitbarek\*\*:**

你好，世界。现在我们发出的是信号还是噪音？我们日日夜夜无休止地写代码，做测试工作，我们追求的事情只有一个：我们是否传递了我们的信号？

**00:00:29 - 各种语言:**

你好。

**00:00:29 - Saron Yitbarek:**

或者我们的信号是不是丢失了？难道我们一直在制造无意义的杂音吗？

**00:00:40:**

我是 Saron Yitbarek，这是代码英雄第二季，来自红帽原创播客。今天的话题将在“翻译”中挖掘。这关于编程语言的故事：它们是怎么来的，如何选择其中一个语言来学习。我们将深入研究如何与机器对话。探究这些语言是如何演化的，以及我们如何用它们来让我们的工作变得更好。

**00:01:08 - Sari:**

Daisy, daisy, 来告诉我你的正确答案。

**00:01:11 - Saron Yitbarek:**

如果你像我一样是一个开发者，你可能也经受着去精通多种语言的压力。最起码你得会 Java，更好一些得懂 Python、Go、JavaScript、Node，甚至能够用 C++ 思考。也许为了增加可信度，可能你还要熟悉古典的 COBOL。甚至你还得学点新东西，比如 Dart。实在太累了。

**00:01:36:**

奇怪的是，我们正在讨论的计算机只讲一种语言：机器语言，以二进制形式飘过的 1 和 0。说到底，我们学习的每个语言都是殊途同归。

不管有多抽象，它都是用来简化我们工作的道具。

**00:02:03:**

这就是我希望你们记住的，故事开始了，从那一刻，那一个经典辉煌的时刻，一个女人说：“你知道么？我是人类，我不想用二进制来讨论思考，我就想用英语去编程。”

**00:02:22:**

在今天可能看来是一个简单的概念。但曾几何时，使用人类的方式与计算机对话这个主意——说轻点是一个笑话，严重点就是亵渎。

**00:02:37:**

代码英雄第二季，将讲述成就我们基础的螺丝钉展开。这一讲的英雄是一个女主人公，如果你想要进一步了解故事，你必须知道她的经历。在此给你们隆重介绍：软件界的第一夫人。

**00:02:58 - 发言人：**

尊敬的先生们和女士们，我非常荣幸的给你们介绍

Grace Mary Hopper  
葛丽丝·玛丽·哈伯准将，谢谢。

**00:03:02 - 葛丽丝·哈伯：**

我曾到加拿大的圭尔夫大学演讲，那时候我必须在多伦多的机场办理入境手续。我把护照递给移民官，他打量了护照和我说：“你是在哪里工作的？”

**00:03:17:**

然后我说，“美国海军。”

**00:03:20:**

她再次深深的打量我一下。然后他说：“你一定是海军里最老的。”

### 00:03:28 - Saron Yitbarek:

她就是 1985 年担任海军少将的葛丽丝·哈伯。在麻省理工学院发表了著名的演讲时，她 79 岁。即使在那时，她也是一个传奇人物。她是独立编程语言之母，实现通过编译器，用人类语言替代数学符号编程这一伟大成就的女性。

### 00:03:51:

她获得了国家技术奖章，她去世后（1992 年），被迫授国家自由勋章。以上记录纯属真实，他们称她为“神奇葛丽丝”。

### 00:04:03 - Claire Evans:

如果有人能够被称为天生的程序员，那么这个人一定是葛丽丝。

### 00:04:06 - Saron Yitbarek:

这是 Claire Evans，一个科技记者和《<sup>Broad Band</sup>宽带》一书的作者，而这本书讲述了科技领域的女性先锋。Evans 描绘了早期的计算机世界，在 1940 年，当葛丽丝·哈伯还是一个年轻的女孩时，她加入了美军的海军预备役部队。当时的电脑有一个小房子那么大。

### 00:04:25 - Claire Evans:

早期的那些程序员，他们就像一个圣职者。他们是一群擅长做枯燥乏味事情的人，因为那会儿没有编译器，没有编程语言，人们是确实是一个比特一个比特地对机器编程。

### 00:04:42:

想要改变这种现实，你真的必须是一个特别的人，而葛丽丝真的就是那种人。

### 00:04:49 - Saron Yitbarek:

现在，她意识到让人类使用机器语言的是一种多么疯狂的限制。就像走路的时候，去告诉你身体的每个原子该怎么做。这是可能的吗？

是，效率呢？不高。

**00:05:06:**

在程序员的意图和计算机的行为之间一定有一条捷径。自从有了算盘，人类一直在将数学思维机械化。一定有一种方法可以在计算机上再次实现这一点。

**00:05:19 - Claire Evans:**

在过去，数学家必须做出所有的步骤。所有这些枯燥的，一步一步的工作都是为了得到一个真实而有趣的答案。然后有了计算机后，它尽可能地承担了这些机械的运算。所以解放了数学家去进行高层面的、更智慧的、面向系统的思考。

**00:05:39:**

只不过，事实并没有如此发展。有了计算机，数学家就变成了程序员。他们又一次被这些枯燥无味的编程所束缚，做着这些一点一滴的繁琐的规则导向的小计算。

**00:05:53:**

因此，我认为，葛丽丝的期望是，她希望计算机能够辅助数学家，然后数学家就可以去更进行宏大的思考，做更伟大的事情，而不至于让他们陷入细节之中。

**00:06:12 - Saron Yitbarek:**

葛丽丝也加入了思想家的行列。我们都知道，帕斯卡在 17 世纪发明了第一台计算器。巴比奇在 19 世纪发明了分析机。这些伟大的杰作是发明家们解放人类的大脑的创作。这些发明让我们处理数据的能力空前绝后。这就是葛丽丝踏入的领域，发明了所有人都认为不可能的东西。

**00:06:42 - Claire Evans:**

我觉得，葛丽丝的想法是，人们应该用人类语言与计算机沟通。这个语言必须独立于机器的，因此不同计算机可以交互操作，这在当时是革命性的。

**00:06:59:**

早期她称这个为“自动化编程”，在程序员和计算机人的群体中，被认为是像太空学员一样。

**00:07:12:**

而那些迟迟不上岸的人们被看成了尼安德特人，这在当时是计算机界的一个巨大的分歧。

**00:07:21 - Saron Yitbarek:**

葛丽丝要说服她的上级跨越这个分歧并不容易。但她认为这是她的责任，她必须尝试。

**00:07:30 - 葛丽丝·哈伯:**

这里总是有一条线，那条线代表了你的老板在那个时候会相信什么。当然，如果你过线了是得不到预算的。所以你对那条线有双重责任：不要踩过它，但也要不断教育你的老板，让你把这条线移得更远，这样下次你就能走得更远。

**00:07:52 - Saron Yitbarek:**

她推动的那个未来，就是我们的现在。几乎你我依赖的每一种语言都归功于她和她的第一个编译器。因此太空学员们活了下来，尼安德特人灭亡了。

**00:08:07:**

写代码不用数字，而是用人类风格的语言是她的观点。你敲下“执行操作 8”，或“关闭 文件 C”的精简操作，才是符合人类习惯的编程。

**00:08:21 - Claire Evans:**

葛丽丝独特地意识到，计算将成为改变世界的事情。但是如果没有人知道如何接触并使用计算机，那么它就不会成为改变世界的事情。因此她想确保它对尽可能多的人开放，尽可能多的人可以使用。

#### **00:08:37 - Claire Evans:**

这就需要编程在可理解性和可读性上做一些妥协。因此最终创造一个编程语言的目标就是给机器提供更多切入点，让它脱离这个神职，让它的开发面向大众和未来新一代。

#### **00:08:59 - Saron Yitbarek:**

我想在这里打断并强调下 **Claire** 的说法：现在我们所已知的编程语言，都来源于科技开放的愿望。这让计算机不再是数学博士们的专属玩具，让编程开发变得更容易。

#### **00:09:14:**

编译器所有工作的本质，是让程序变得更好理解，更有人性。

#### **00:09:21:**

**Claire** 有一个猜测，为什么葛丽丝是做出改变的那个人，这与她在二战期间的工作有关。

#### **00:09:29 - Claire Evans:**

她的工作是解决扫雷问题、弹道问题和海洋学问题。她用很多不同的、交叉的学科来模拟战争里的所有的暴力、混乱和现实的灾难，并将他们转化成在 **Mark I** 计算机上运行的程序。

#### **00:09:45:**

她知道如何在语言之间去做翻译转换。我的意思不是说计算机语言，是人类语言。她知道如何倾听一个提出复杂问题的人的意见，并尝试理解问题的背景，其信息和所涉及的专业规律，然后将这些转为计算机可以理解的程序。

#### **00:10:06:**

从这个角度看她如同早期的编译器。就像一个人类版本的编译器，因为她知道你必须理解他们才能满足他们的需求。

**00:10:17 - Saron Yitbarek:**

编译器干的事情就是一个解释和理解。我觉得我们应该在学习新语言，或想知道为什么有一些东西根本不能编译的时候牢记这个理念。编译器的工作就是满足你使用生活中说的语言来编程。

**00:10:32:**

葛丽丝知道人类一旦可以学会讲编程语言，而编译器可以将我们的意图转换为机器语言，就像为洪水打开了大门一样。

**00:10:43 - Claire Evans:**

她知道如果计算机太孤立和太具体，就不会发展为一个产业，从而成为改变世界的力量。也就是说计算机的从业者，可以让提出问题的人跟机器直接沟通。

**00:11:00:**

因此她真的是一个善于人类语言翻译的专家，至少我是这么想的。这使她有独一无二的机会，去思考和创建编程语言。

**00:11:15 - Saron Yitbarek:**

葛丽丝对英文式数据处理语言的研究最终演变成了 COBOL，它在某种意味上很不错。因为它不浮华，很适合商务用途，葛丽丝·哈伯也是这样的人。

**00:11:28:**

从某种角度看，她给了我们一个很像她的语言。像葛丽丝一样，COBOL 语言也很长寿，它现在已经 60 多了。

**00:11:50:**



葛丽丝的编译器就像一个巴别鱼，在程序员和机器之间充当翻译，不过它们翻译的都是高度概括性的语言。

**00:12:03:**

然后，几十年之后，另一个重要的思潮在语言界兴起。想象下画面：自由软件社区在 1980 年出现，但是 Unix 的替代品 GNU 被开发出来后，却没有任何自由开放的编译器随之出现。

**00:12:22:**

为了让 GNU 给我们提供一个真正的开源 UNIX 替代品，为了让编程语言在开源世界蓬勃发展，社区需要找来一个格蕾丝·哈伯 —— 我们需要一个开源编译器。

**00:12:38:**

这是 Langdon White，红帽的平台架构师，来讲讲他对这个事情的理解。

**00:12:45 - Langdon White:**

在 80 年代，你要买一个编译器很轻松就要花上上万块钱。费用是最大的问题，我没有多余的钱去给自己买编译器。再一个事实是，我必须为所有我想要的目标平台买一个对应的编译器。那个时代大部分是 Unix 平台，但是细节和风格各不相同。

**00:13:06:**

因此你就不能买一个，你需要为不同的架构，不同的供应商购买多个编译器。

**00:13:14 - Saron Yitbarek:**

Langdon 指出这不仅仅是成本问题，在一些情况下，对编码工作也带来了问题。

**00:13:21 - Langdon White:**

大家都没有意识到，你如何用很特殊的方式来组织你的代码是很重要的。做嵌套 **for** 循环或者做嵌套 **while** 循环之类的事情可能是可以的，这取决于编译器。

**00:13:38:**

因此，大家都应该明白，如果你不知道编译是如何优化你的代码的，就很难知道如何写出最优的代码。

**00:13:49 - Saron Yitbarek:**

必须要提的是，我们需要一个免费的、可获得的、可值得信赖的编译器。这就是 **GNU C 语言编译器：GCC**。它横空出世在 1987 年，它是格蕾丝·哈伯的编译器革命和自由软件运动的结合。

**00:14:12:**

它是使编译更标准化，从而让所有人可以编译别人写的代码。我们编程语言的丰富性要归功于它。

**00:14:22 - Carol Willing:**

GCC 是开放的，可以说将编程语言推向一个更高的层次。

**00:14:29 - Saron Yitbarek:**

这是 Jupiter 项目成员 Carol Willing，她是 Python 软件基金会的前理事。

**00:14:35 - Carol Willing:**

人们开始意识到，专有的语言会在特定时间内被服务于一个目的，但并不能得到开发者社区的认可和热爱。因为如果你是一个开发者，你希望学习最常用的，以及未来最流行的东西。

**00:14:59:**

我没必要去发展一种将我锁定在一种技术上的技能。我想 Python 成功的一个原因是因为它是开源的，它有非常简洁的语法。

**00:15:11:**

它的特点就是允许人们用常见的方法，快速高效地解决问题，也让大家合作解决问题。我想这就是好的程序和代码库的标志：如果你可以用最小的工作量完成你的工作，并且与他人分享，这是确实很棒的事情。

**00:15:35 - Saron Yitbarek:**

这么多年过去了，GCC 逐渐的支持了 Java、C++、Ada 和 Fortran 语言，我还可以继续说说它的进步。

**00:15:43 - Carol Willing:**

通过像 GCC 这样的通用底层接口，人们可以根据自己的特殊需求来定制语言。例如，在 Python 的世界里，有大量的库，甚至具体到科学 Python 世界里，我们有 numpy，还有 scikit-image、scikit-learn 这样的东西。

**00:16:08:**

每个库都是为一个特定目的而工作。因此，我们也看到了生物信息学和自然语言处理之类的东西。而人们可以在一个共同的基础上，做出很多不同的事情。而且可以把它们放到编程语言或库里，使他们能够在他们特定的行业或领域中优化问题的解决。

**00:16:42 - Saron Yitbarek:**

因此，这就是编译器技术一头撞进开源运动的结果吧？随着时间的推移，这种不同技术的碰撞，爆炸般地创造了一个新的、被社区支持的语言，大家都可以学习和完善它。

**00:16:58:**

现在有成百上千的编程语言存活着。

**00:17:03 - Carol Willing:**

随着开源软件越来越流行和被广泛接受，我们看到了编程语言的大量激增。现在有大量围绕着手机技术的编程语言，不同的编程语言也让游戏开发更加简单快速便捷。各种用途的语言，如 **Python** 和 **Ruby**，它们算是现代网页开发和交付网页应用和网站的基础。

#### **00:17:34 - Saron Yitbarek:**

这个队伍还会继续壮大。是的，我们建造的这个巴别塔在未来会更加拥挤。但是你可以把它当作一个聚宝盆，一个语言的盛宴。下面我们将会帮你们梳理这个盛宴。

#### **00:17:55:**

现在我们已经知道编程语言泛滥的原因了。但是这些对我们有什么具体的意义？我们如何选择对我们重要的语言呢？这是个大问题，因此我们找了一些人帮忙：**Clive Thompson**，是最好的作家之一，他能让科技世界变得更有意义。他是《连线》的专栏记者，《纽约时报》杂志的特约撰稿人，他现在正在写一本关于计算机程序员心理学的书。

#### **00:18:24:**

当我们挑选我们想要学习的编程语言时，我们需要知道我们到底想要什么。

#### **00:18:31:**

这是我和 **Clive** 反复讨论得出的结论。

#### **00:18:35:**

当我作为一个开发新人第一次入手的时候，我就说：“让我选一个最好的编程语言，然后掌握并熟练运用它，不就完事了么。”

#### **00:18:44:**

不过事情不会这么简单，否则为什么有那么多的编程语言呢？

#### **00:18:49 - Clive Thompson:**

每个语言都有那么点它的特长。因此通常来说，有人创造一个新语言是因为现有的语言满足不了他们的需求。**JavaScript** 就是一个典型的例子。

**00:19:03:**

Netscape

网景公司曾经在 90 年代中开发了一款浏览器，所有的网站管理员想做一些更具交互性的事情。他们希望有一种方法，使其可以在网站上运行一些脚本。

**00:19:16:**

当然这些需求都提给了网景。然后他们说：“好吧，现在我们没有可以做到这点的办法，我们需要一个可以集成到我们浏览器的脚本语言。”

**00:19:25:**

于是他们雇佣了 **Brendan Eich**，一个被认为很资深的家伙。他当时 32 岁，其他人 21 岁的样子。

**00:19:32 - Saron Yitbarek:**

这在开发者圈里就很资深了

**00:19:35 - Clive Thompson:**

他们给他 10 天时间开发 **JavaScript**。因此他真的就 10 天没有睡觉，然后疯狂地捣鼓出一个脚本语言。它看起来有点挫，有点糟，但是可以运行。它允许人们做一些简单的事情，它还有按钮，对话框，弹框和其他东西。

**00:19:54:**

这就是造一个编程语言的例子，用来做在当时没有办法完成的事情。

**00:20:01 - Saron Yitbarek:**

是不是很有意思。

**00:20:03 - Clive Thompson:**

这就是为什么存在这么多编程语言，人们不断进取，去做别人做不到的事。

**00:20:11 - Saron Yitbarek:**

这也是为什么我对开发者和编程语言之间的关系很感兴趣，我们对这些工具有很强的依赖性。为什么呢？

**00:20:22 - Clive Thompson:**

有几个原因。一个是有时你学习你的第一个编程语言完全是一场意外，有点像你的初恋。

**00:20:30:**

我觉得不同性格的人，也会匹配不同类型的编程语言。例如 Facebook 是用 PHP 设计的。PHP 是有点像毛球，它很不规则，它的成长断断续续，Facebook 也有点这种感觉。

**00:20:49:**

与此相反，来自谷歌的伙计决定：“我们想要一个超高性能的语言，因为在谷歌我们的东西都高速运行着，维护着万亿级的用户吞吐。”

**00:21:02:**

因此他们决定开发一个 Go 语言，Go 非常适合高性能计算。

**00:21:08 - Saron Yitbarek:**

让我们深入一点。作为一个开发者，我有自己的个性，我是不是对一部分编程语言有天然喜欢呢？或者我工作用的编程语言可能会影响我的个性？

**00:21:25 - Clive Thompson:**

两者肯定都存在。但我确实认为当你遇到你喜欢的编程语言的时候，有一种强烈的共鸣感。你上计算机课程时学习了必修语言：他们都教 Java，有时会多教一门 JavaScript 或 Python。

**00:21:46:**

问题是，你被强制要求，所以你学了它。但当你有选择余地时会怎么样呢？这就是你真正看到一个人的那种情感或心理风格是如何倾向于一门语言的地方，因为我和一个人谈过，他试着学了一堆 JavaScript。

**00:22:03:**

这是个看起来有点丑的语言，它简直是花括号的噩梦。所以他试了又试，试了又试，失败了又失败。直到有一天他看到朋友在用 Python。在他看起来，这是多么的纯净和美妙的语言。因为他是个作家，而 Python 被认为是一个书写型的编程语言，缩进使一切都易于阅读。

**00:22:28 - Clive Thompson:**

它和他一拍即合，主要在于 Python 的工作方式和外观，漂亮的排版打动了它。

**00:22:39 - Saron Yitbarek:**

和 Clive 的交流让我意识到，有一些编程语言是硬塞给我们的。我说的是那些古老的编程语言，已经类似于拉丁语了。

**00:22:53:**

其实我们也有很多选择，去选择适合我们个性的编程语言。如果你想知道最新的东西有哪些，你去问问开发者们周末在用什么就知道了。

**00:23:05:**

下面是我们对话的更多内容：

**00:23:08 - Clive Thompson:**

当我问人们：“你闲暇的时候做什么？”回答肯定是些稀奇古怪的东西。我想实际上其中一个好的，最值得称颂的开发者的特点，就是他们是富有好奇心。

**00:23:22:**

我知道有人会说，“我要学习 Erlang 了，就是为了好玩。”

**00:23:26 - Saron Yitbarek:**

我尝试想象人们在周末开发这些项目，项目不是最重要的，他们在学习工具，编程语言才是重要的事情。这才是他们真的想要的状态。

**00:23:41 - Clive Thompson:**

确切的说，你将看到为什么大家不停的重复迭代开发这些日历和待办事项，因为这是一个快速的学习方法。但需要指出的是，编程语言做了什么以及如何工作都与我们没关系，我只需要尽可能的构建我的东西。

**00:23:56:**

他们想知道用那样的编程语言思考是什么感觉。是不是会感觉很轻松、刺激、游刃有余？我目前的使用的语言没这个感觉，是不是入门之后一切事情都变得简单了？

**00:24:13 - Clive Thompson:**

很有可能你遇到一个新语言后会非常兴奋，想起来都很兴奋。

**00:24:20 - Saron Yitbarek:**

我是一个 Ruby 从业者，作为 Ruby on Rails 开发者我非常自豪，我想大概是两年前，另一个非常著名的 Ruby 开发者，Justin Serrals 做了一个非常好的演讲，倡导编程语言从实用角度出发，没有必要那么性感。

**00:24:41:**



他的观点是，**Ruby** 之所以是一个令人兴奋的编程语言的原因在于它很新，而且差不多几年前都已经得到印证了，它已经不需要更新什么了。它已经是一个稳定、成熟的编程语言了。不过因为它的成熟，开发者对它也就不那么兴奋，我们逐步把目光转移到新的花样来了。

**00:25:05:**

从某个意义上说，很有可能我们的好奇心会扼杀了一个编程语言，或者让它变得庸俗，而这与编程语言本身的好坏没有关系。

**00:25:18 - Clive Thompson:**

我认为这是完全正确的。事实上，你看到的开发者这个极度好奇心和自学欲望的特点，会让他们不断地寻找新鲜事物。并用这种语言去复制其他语言已经基本做得很好的东西。

**00:25:37 - Saron Yitbarek:**

是的。

**00:25:37 - Clive Thompson:**

就是这样，好奇心有好处，同时也是一个陷阱。

**00:25:43 - Saron Yitbarek:**

是的，说的很好。

**00:25:49:**

有时我们的好奇心可能是一个陷阱，但是它也是编程语言革命的动力。开发每个新编程语言的时候他们会说，“有没有别的可能？”它们出现的原因在于，开发人员想要做不一样的事情。

**00:26:06:**

我们想要全新的表达方式。

**00:26:08 - 葛丽丝·哈伯:**

我向你们承诺一些事情。

**00:26:11 - Saron Yitbarek:**

我想葛丽丝·哈伯想在这儿最后讲两句。

**00:26:15 - 葛丽丝·哈伯:**

在未来 12 个月里，你们谁要是说我们一直都是这么做的，我将会瞬间实质化在你的旁边，然后我 24 小时围绕着你。我倒是要看看我能不能引起你的注意。我们不能再相信那句话，这是很危险的。

**00:26:34 - Saron Yitbarek:**

葛丽丝的故事和第一款编译器提醒我们，只要能可以找到表达方法，我们总有更好的办法做事情。

**00:26:43:**

不管编程语言有多抽象，不管我们在机器的 1 和 0 高位还是低位浮动，我们都需要确保它是一个明智选择。我们选择一个语言，是让它帮助我们的想法更容易表达出来。

**00:27:03:**

如果你想更深一步学习编程语言和编译器，你不会很孤独，我们收集了一些很有用的文档帮你深入学习。敬请在我们节目下面留言。请查看 [redhat.com/commandlineheroes](https://redhat.com/commandlineheroes)

**00:27:20:**

下一期节目，我们将关注开源贡献者的艰辛之路。那些维护者的真实奋斗生活是什么样的？我们如何提出第一个拉取请求？

**00:27:32:**

我们将带你们简单认识开源贡献。

**00:27:39:**

代码英雄是红帽的原创播客。你可以在苹果播客、谷歌播客以及其他可能的地方免费收听这个节目。

**00:27:48:**

我是 Saron Yitbarek，坚持编程，下期再见。

## 什么是 **LCTT SIG** 和 **LCTT LCRH SIG**

Special Interest Group  
LCTT SIG 是 LCTT 特别兴趣小组，LCTT SIG 是针对特定领域、特定内容的翻译小组，翻译组成员将遵循 LCTT 流程和规范，参与翻译，并获得相应的奖励。LCRH SIG 是 LCTT 联合红帽（Red Hat）发起的 SIG，当前专注任务是《代码英雄》系列播客的脚本汉化，已有数十位贡献者加入。敬请每周三、周五期待经过我们精心翻译、校对和发布的译文。

欢迎[加入 LCRH SIG](#)一同参与贡献，并领取红帽（Red Hat）和我们联合颁发的专属贡献者证书。

---

via: <https://www.redhat.com/en/command-line-heroes/season-2/hello-world>

作者: [Red Hat](#) 选题: [bestony](#) 译者: [guevaraya](#) 校对: [acyanbird](#), [wxy](#)

本文由 [LCRH](#) 原创编译，[Linux中国](#) 荣誉推出