

Why do some people like programming?

Marcus Geduld, 30 years of coding.

- I like creating something out of nothing. That's not literally what you do when you're programming, because there's existing hardware and software that serves as a foundation for your work, but it sure feels that way. Someone has an idea and you build it from the ground up. When you begin, there's just an empty text editor. When you're done, there's a (hopefully) working program.
- 我喜欢从无到有的创造事物。虽然编程时并不完全是这样，因为你需要一些基本的硬件和软件来为你服务，但实际上你的感觉就是这样。你可以把别人一个想法从头开始构建出来。当你开始的时候，只有一个空白的记事本，当你完成的时候，却有一个可以运行（但愿如此）的程序。
- I like building things people use. It's amazing to type up some code, press a button, and suddenly thousands of people on the Internet are playing with it.
- 我喜欢创造出让别人使用的东西。敲一些代码，点几个按钮，忽然之间数以千计的人在互联网上开始使用它，这是多么让人惊奇的事。
- I like playing God. Programming allows you to build little worlds and then play with them, making adjustments and watching the effects. It's like owning a toy planet and saying, "I'm going to make it rain, today. Oh, look! All the little people have opened umbrellas!" This playing-God aspect of programming makes it similar to writing novels, painting paintings, or directing plays.
- 我喜欢扮演上帝。你可以通过编程创建一个小型的世界并参与其中，做出调整并观察效果。就像你拥有一个玩具星球，你说：『今天下点雨吧，哦，看！所有的小人都打开了雨伞！』。编程的这种类似于扮演上帝的体验使得它与写小说、绘画以及导演戏剧有相似之处。
- I like working within systems that demand precision. This is exactly what some people hate about programming, but it thrills me. A misplaced semicolon or the smallest typo can be disastrous. This keeps me on my toes. It's like being the butler on "Downton Abbey" or "Upstairs, Downstairs." Everything must be just so. Some people like precision; others like being about to say, "I can't describe it, but you know what I mean..." I'm the former type.
- 我喜欢和需求精确的系统打交道。这却是一些人痛恨编程的原因，但却令我感到兴奋。一个错位的分号或一个很小的错字都会导致灾难。这让我一直保持警惕。就像『唐顿庄园』或『楼上楼下』中的男管家，所有的事情都必须这样做。有些人喜欢精确；有的人喜欢这样说，“我不好描述，但你知道我的意思……”，而我显然是前者。

- I like solving puzzles. If you want to see if programming is for you, try out this puzzle book: [The Little Schemer – 4th Edition: Daniel P. Friedman, Matthias Felleisen, Duane Bibby, Gerald J. Sussman: 9780262560993: Amazon.com: Books](#)
- 我喜欢猜谜游戏。如果你想知道你是否适合编程，可以看一下这本猜谜书：[『Little Schemer』第四版](#)
- I like research. Programming tends to involve much googling and reading through documentation.
- 我喜欢钻研。编程常常需要做大量的google以及阅读文档。
- I like experimenting. There's a large component of trial-and-error.
- 我喜欢实验。编程中的一大部分都是在试错。
- I like writing poetry, which is very similar to programming. Both the poet and the programmer are obsessed with words, obsessed with formal rules, obsessed with seeing how far they can push those rules, and obsessed with expressiveness. Programmers often talk about how expressive a particular programming language is. They mean something very similar to what a poet means when he tries to come up with expressive wording.
- 我喜欢写诗，这和编程很相似。诗和程序都是沉迷于词，沉迷于规则，沉迷于它们能将这规则发挥到什么程度，沉迷于表达。程序员常谈及某一种编程语言的表达力。这和一个诗人试图提高语言的表达力是一致的。

Both poetry and code can be exquisitely beautiful -- and in very similar ways. People are often surprised that I'm a programmer who also directs Shakespeare plays, but I've met lots of programmers who are into Shakespeare, crossword puzzles, scrabble, and so on. [Joshua Engel](#) is another Quora user who writes code and directs Shakespeare plays.

诗和代码都可以相同的方式达到精致的美。人们常惊奇于我是一个程序员同时又在导演莎士比亚的戏剧，但我遇见果很多喜欢莎士比亚、填词游戏、拼字游戏等等的程序员。[Joshua Engel](#) 也是一个既写代码又导演莎士比亚的Quora用户。

- I like communicating. Most good programmers will tell you that code is first-and-foremost meant to be read by people -- even to the extent that we'll sometimes write it in a way that is inefficient but easy-to-read.
- 我喜欢交流。好的程序员会告诉你，代码首要的是被人来读，甚至于我们有时宁肯牺牲效率也要易读。
- "There are only two hard things in Computer Science: cache invalidation and naming things." -- Phil Karlton. I love the hard work of "naming things." Programmers generally have to come up with short words or phrases that label parts of the systems they're writing. It's crucial that these names be clear and apt.
- 『计算机科学有两大难题：缓存失效和命名问题。』Phil Karlton曾这样说过。我喜欢『命名问题』。程序员通常需要用单词和短语来标识他们编写的系统。这些名字必须简洁而恰当。

Why? Because if you name something `cButton`, the next guy who has to work on your code may be befuddled. If you'd called it `closeButton`, he would have instantly known what you were referring to. Sometimes "he" is me a few weeks (or a year) later, when I'm reading my own code. It's embarrassing to come across `cButton` and think, "What did I mean by that?"

为什么？比如你命名了一个东西叫`cButton`，下一个接管你代码的人就会很困惑。但如果你命名为`closeButton`，他就会马上明白你的意思。有时候这个『他』就是几周（或一年）后在读代码的我自己。当遇见`cButton`时，不免尴尬的想，这个到底是啥意思？

Last week, I was modifying someone else's code. It was for a web page with sections. Each section had a logo at the top. The original programmer had referred to those logos internally as "headers," e.g. "header1" and "header2." I didn't notice that, and so I named something within one of the sections "header." When I later looked over the code, I got totally confused between his headers and mine.

上周，我修改一份别人的代码。这是一个使用section的web页面。每个section的顶部都有一个logo。原来的程序员把这个logo命名为『headers』，就像『header1』和『header2』这样。我没有注意到这一点，我把section内一些内容定义为『header』。后来我在看代码的时候，我已经分不清到底是他的header还是我的header，彻底晕了。

Then I thought it over and realized that his "headers" were always graphical logos and mine were text. So I renamed his "logo1" and "logo2" and mine "title."

然后我想了一下，意识到他的『header』总是指代logo图片，而我的是文本，于是我把他的header重命名为『logo1』和『logo2』，我的命名为『title』。

This is just one small example of the sort of naming issues that constantly crop up. You either enjoy this sort of thing or you don't. I do.

这只是经常发生的命名类问题的一个小例子。不管你是否喜欢这一类问题，我是喜欢的。

- I like learning. Like sharks, programmers die if they stop moving. Because technology changes so fast, being a programmer means that "school" never stops. Even though I've been coding for years, I must constantly read programming books, follow blogs, and so on. There's no coasting!
- 我喜欢学习。就像鲨鱼一样，程序员止步则死。技术飞速发展，程序员要活到老学到老。虽然我写了几十年代码，我依然在读编程的书，订阅技术博客等等，不计成本！

I got into the game as a Flash programmer, which was lucrative for about ten years. Now Flash seems to be on its way out, so it's back to the books! But even while I was mostly coding Actionscript (Flash's language), I needed constant training, because that language went through many versions, some as different from each other as Spanish is from Portuguese.

我曾经是Flash游戏程序员，这在十年前很风光。如今Flash已经过时了，那就重新回到书中吧！即使在我使用Actionscript（Flash编程语言）写代码的时候，我也需要不断的训练，因为语言本身发展了很多的版本，有些版本间的区别就像西班牙语和葡萄牙语的差别。

There are many good programming books and courses, but you can't really learn to code by instruction. That will get you started, but the only real way to learn is to write code, fail, analyse the failure, and learn from it. So you must enjoy being an autodidact. I do.

现在有许多编程书籍和课程，但你不能仅凭教导真正学会编程。这只是让你入门。写代码，失败，分析错误并从中学习是学会编程的不二法门。所以你必须喜爱自学，我就是这样。

- I like being a detective. Maybe 60% of programming is debugging -- figuring out how something works. That often means a ton of sleuth work. Sometimes you have to pick an entire program apart and put it back together again.
- 我喜欢做一个侦探。编程中60%的时间都是调试，以寻求正常运行的方法。这往往意味着大量的检测工作。你时常需要拿掉一大块程序然后又把它放回去。
- I like solitary work. Programming allows me to do lots of that.
- 我喜欢独自工作。编程总是这样的。
- I like collaborating. Nowadays, few programmers work completely alone. Most are part of a team, and they spend part of the week doing close work with others and part in isolation. I feel a strong need for both sorts of work, and I like alternating between the two.
- 我喜欢合作。现今，很少有人是完全独自编程的。多数人都是团队中的一员，每周的工作，部分是与他人紧密合作，部分是独立在做。两种方式我都需要，而且我喜欢这样的切换。