

**Jeff Maxim**

**Nov 7, 2015 at 4:05pm** *day ago*

Hey, I'm Jeff, and I teach CS50AP in NYC. Check me out here: <http://cs50apnyc.com/> and here <http://jeffmaxim.herokuapp.com/>

version of why to learn C:

A programmer who has learned a lower-level programming language like C has a deeper understanding of foundational CS concepts like pointers, data types, passing values by reference, etc. than a programmer who's only learned more modern, high-level languages.

### **More details about why to learn C:**

I think there's three main reasons to learn C.

First, C has been around for almost 30 years, and there's tons of resources available. There's countless textbooks, online resources, and a bunch of example source code to check out. It's very stable. Whereas a more high-level, modern language like python is constantly changing, and new updates are sometimes not fully adopted by the whole python community (python 2 vs. python 3, for example), C doesn't change much nowadays. As a result, C is somewhat of a lingua franca of programming languages ([https://en.wikipedia.org/wiki/Lingua\\_franca](https://en.wikipedia.org/wiki/Lingua_franca)). ;

Second, foundational concepts in C show up in basically every programming language that exists today. Concepts such as argc and argv, loops, and variable types are the foundations upon which many more modern, high-level languages are built.

Third, C code is pretty close to machine code, and writing C code helps you understand how code interacts with the machine. Bytes, bits, pointers, compilers, and optimization techniques are things that a C programmer must understand (wait until you get to the spell-checker problem set!). There is tremendous value in knowing what goes on under-the-hood in a programming language, and C forces you to acquire that in depth understanding of how a programming language interacts with a machine. There's plenty of bad Ruby, Python, and JavaScript code out there, code that's syntactically correct and runs, but which is egregiously bloated and slow. It's harder to understand what goes on under-the-hood with a higher-level language. When something goes wrong with your python code, it's easier to deduce what the problem is when your programming knowledge is built on an in-depth knowledge of what goes on at the machine level.

A fourth, bonus point, is that knowing C and writing it well gives you super bragging rights. When I tell experienced programmers that I'm teaching high school computer science in C, they're impressed. It's hard, so if you master it, everything else seems easier.

I got most of my ideas for this post from here: <http://www.cprogramming.com/whyc.html>

The counter argument is that higher-level languages are much more accessible, and it's easier to get a cool program up and running more quickly. Learning an old language has you stuck in the past.

Anyway, there's definitely two sides to the argument. As a dude who's self-taught himself to code, and who first learned in C, I've found that the foundational understanding of CS that I picked up from my C programming has helped me tremendously with my understanding and use of more modern languages.