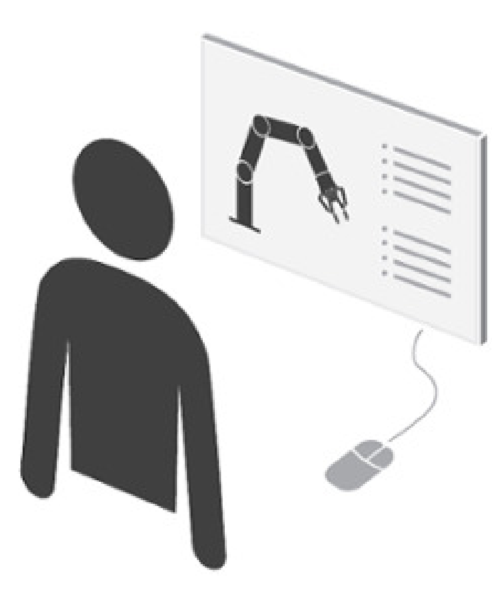
What is the Best Programming Language for Robotics?

by Alex Owen-Hill on Mar 14, 2016 7:00:00 AM

Tweet Share Like 145 Share

It's a question that a lot of new roboticists will ask at least once in their career. Unfortunately, it's also a question which doesn't have a simple answer. In this post, we'll look at the top 10 most popular programming languages used in robotics. We'll discuss their strengths and weaknesses, as well as reasons for and against using them.

It is actually a very reasonable question. After all, what's the point of investing a lot of time and effort in learning a new programming language, if it turns out you're never going to use it? If you are a new roboticists, you want to learn the programming languages which are actually going to be useful for your career.



Why "It Depends" is a Useless Answer

Unfortunately, you will never get a simple answer if you asked "What's the best programming language for robotics?" to a whole roomful of robotics professionals (or on forums like Stack Overflow, Quora, Trossen, Reddit or Research Gate).

Electronic engineers will give a different answer from industrial robotic technicians. Computer vision programmers will give a different answer than cognitive roboticists. And everyone would disagree as to what is "the best programming language". In the end, the answer which most people would all agree with is "it depends." This is a pretty useless answer for the new roboticist who is trying to decide which language to learn first. Even if this is the most realistic answer, because it does depend on what type of application you want to develop and what system you are using.

Which Programming Language Should I Learn First?

It's probably better to ask, which programming language is the one you should start learning first? You will still get differing opinions, but a lot of roboticists can agree on the key languages.

As we discussed in our post about essential robotic skills, the most important thing for roboticists is to develop "The Programming Mindset" rather than to be proficient in one specific language. In many ways, it doesn't really matter which programming language you learn first. Each language that you learn develops your proficiency with the programming mindset and makes it easier to learn any new language whenever it's required.

Top 10 Popular Programming Languages in Robotics

There are over 1500 programming languages in the world, which is far too many to learn. Here are the ten most popular programming languages in robotics at the moment. If your favorite language isn't on the list, please tell everyone about it in the comments! Each language has different advantages for robotics. The way I have ordered them is only partly in order of importance from least to most valuable.

10. BASIC / Pascal

BASIC and Pascal were two of the first programming languages that I ever learned. However, that's not why I've included them here. They are the basis for several of the industrial robot languages, described below. BASIC was designed for beginners (it stands for Beginners All-Purpose Symbolic Instruction Code), which makes it a pretty simple language to start with. Pascal was designed to encourage good programming practices and also introduces constructs like pointers, which makes it a good "stepping stone" from BASIC to a more involved language. These days, both languages are a bit outdated to be good for "everyday use". However, it can be useful to learn them if you're going to be doing a lot of low level coding or you want to become familiar with other industrial robot languages.

9. Industrial Robot Languages

Almost every robot manufacturer has developed their own proprietary robot programming language, which has been one of the problems in industrial robotics. You can become familiar with several of them by learning Pascal. However, you are still going to have to learn a new language every time you start using a new robot.

ABB has its RAPID programming language. Kuka has KRL (Kuka Robot Language). Comau uses PDL2, Yaskawa uses INFORM and Kawasaki uses AS. Then, Fanuc robots use Karel, Stäubli robots use VAL3 and Universal Robots use URScript.

In recent years, programming options like ROS Industrial have started to provide more standardized options for programmers. However, if you are a technician, you are still more likely to have to use the manufacturer's language.

8. LISP

LISP is the world's second oldest programming language (FORTRAN is older, but only by one year). It is not as widely used as many of the other programming languages on this list; however, it is still quite important within Artificial Intelligence programming. Parts of ROS are written in LISP, although you don't need to know it to use ROS.

7. Hardware Description Languages (HDLs)

Hardware Description Languages are basically a programming way of describing electronics. These languages are quite familiar to some roboticists, because they are used to program Field Programmable Gate Arrays (FPGAs). FPGAs allow you to develop electronic hardware without having to actually produce a silicon chip, which makes them a quicker and easier option for some development. If you don't prototype electronics, you may never use HDLs. Even so, it is important to know that they exist, as they are quite different from other programming languages. For one thing, all operations are carried out in parallel, rather than sequentially as with processor based languages.

6. Assembly

Assembly allows you to program at "the level of ones and zeros". This is programming at the lowest level (more or less). In the recent past, most low level electronics required programming in Assembly. With the rise of Arduino and other such microcontrollers, you can now program easily at this level using C/C++, which means that Assembly is probably going to become less necessary for most roboticists.

5. MATLAB

MATLAB, and its open source relatives, such as Octave, is very popular with some robotic engineers for analyzing data and developing control systems. There is also a very popular Robotics Toolbox for MATLAB. I know people who have developed entire robotics systems using MATLAB alone. If you want to analyze data, produce advanced graphs or implement control systems, you will probably want to learn MATLAB.

4. C#/.NET

C# is a proprietary programming language provided by Microsoft. I include C#/.NET here largely because of the Microsoft Robotics Developer Studio, which uses it as its primary language. If you are going to use this system, you're probably going to have to use C#. However, learning C/C++ first might be a good option for long term development of your coding skills.

3. Java

As an electronics engineer, I am always surprised that some computer science degrees teach Java to students as their first programming language. Java "hides" the underlying memory functionality from the programmer, which makes it easier to program than, say, C, but also this means that you have less of an understanding of what it's actually doing with your code. If you come to robotics from a computer science background (and many people do, especially in research) you will probably already have learned Java. Like C# and MATLAB, Java is an interpretive language, which means that it is not compiled into machine code. Rather, the Java Virtual Machine interprets the instructions at runtime. The theory for using Java is that you can use the same code on many different machines, thanks to the Java Virtual Machine. In practice, this doesn't always work out and can sometimes cause code to run slowly. However, Java is quite popular in some parts of robotics, so you might need it.

2. Python

There has been a huge resurgence of Python in recent years especially in robotics. One of the reasons for this is probably that Python (and C++) are the two main programming languages found in ROS. Like Java, it is an interpretive language. Unlike Java, the prime focus of the language is ease of use. Many people agree that it achieves this very well. Python dispenses with a lot of the usual things which take up time in programming, such as defining and casting variable types. Also, there are a huge number of free libraries for it, which means you don't have to "reinvent the wheel" when you need to implement some basic functionality. And since it allows simple bindings with C/C++ code, this means that performance heavy parts of the code can be implemented in these languages to avoid performance loss. As more electronics start to support Python "out-of-the-box" (as with Raspberry Pi), we are likely to see a lot more Python in robotics.

1. C/C++

Finally, we reach the Number 1 programming language in robotics! Many people agree that C and C++ are a good starting point for new roboticists. Why? Because a lot of hardware libraries use these languages. They allow interaction with low level hardware, allow for real time performance and are very mature programming languages. These days, you'll probably use C++ more than C, because the language has much more functionality. C++ is basically an extension of C. It can be useful to learn at least a little bit of C first, so that you can recognize it when you find a hardware library written in C. C/C++ are not as simple to use as, say, Python or MATLAB. It can take quite a lot longer to implement the same functionality using C and it will require many more lines of code. However, as robotics is very dependent on real time performance, C and C++ are probably the closest thing that we roboticists have to "a standard language".

In Which Order Should You Learn Them?

Just because I've listed these ten doesn't mean that you have to learn all of them, or indeed any of them. The most important thing is to find a language that feels natural for you and fits with your robotic hardware. You want a language which allows you to develop programs quickly and easily, so that you can focus more on developing functionality. For this reason, I would recommend learning Python first. This is just my own personal opinion. If another language makes more sense for you, then learn that instead. However, Python is an amazingly straightforward language to learn and hugely powerful thanks to the many, easily accessible libraries. I have heard many accounts from (already experienced) programmers who learned Python in a couple of days and were immediately converted to it for almost all of their programming needs. As one guy said, "*I can produce usable code in Python as fast as I can type*". After you've gotten reasonably proficient using Python, I would personally recommend learning C, followed by C++. You will need it to interface with a huge majority of robotic hardware drivers.

Have we left out your favorite programming language? Which programming language did you learn first? Which do you use most often when programming robots? ***Tell us in the comments below or join the discussion on LinkedIn, Twitter or Facebook.***