POPULAR PROGRAMMING LANGUAGES FOR IOT DEVELOPMENT

veryday life. Today, the Internet of Things (IoT) is very much a reality with smart thermostats, connected cars, and smart home hubs like Amazon Echo (that uses multiple languages like Node.js, Java, and Python).

So how do they work? What kind of coding do they involve?

If you bring it down to the basics, “smart things” are using a lot of the same languages that are used by applications on your personal computers and mobile devices.

According to the survey of developers conducted by the Eclipse Foundation, the top four languages for building IoT solutions are as follows:

- Java
- C
- JavaScript
- Python

These languages are more or less the same when it comes to desktop apps, mobile apps, and servers. So it might seem like there’s no difference other than smart objects are like little computers. Although this is sort of true, there are some significant differences when it comes to all the things that make up IoT.

There are three major sections in an IoT architectural environment. These are the sensors that generate the data, the local gateways or hubs that organize it, and the geographically distant and centralized servers where all the data ends up.

If it’s a basic sensor, it’s probably using C as it can work directly with the RAM. For the rest, developers will be able to pick and choose the language that best suits them and the build.

As the hubs will be like a small console or a smartphone, they will be using a standard OS. So communication via a command line will feel the same as what developers have grown accustomed to.

How Do you Choose the Right Language for your IoT Project?

Choosing a programming language for your IoT build will be the same as any development project. This is because its behavior won’t be any different from a laptop, tablet, or server. When you’re dealing with sensors and hubs with some form of microservice architecture, the data will be pushed into a standard database.

The IoT space is already dominated by the usual popular languages, but there are plenty more used by developers to build interesting smart things.

Usually, IoT is a polyglot effort, so you're not going to see a heavy reliance on a single language. So let's take a look at some of the major players and some of the not so popular ones.

1. Assembly

Although it's not the most popular language on the list, Assembly is a great option if you want to keep your IoT applications compact.

It's a low-level programming language, so don't expect to do much with it as its capabilities are minimal.

2. B#

B# was specifically developed for small applications, so you can use this language on multiple platforms using an Embedded Virtual Machine (EVM) that supports B#.

If you're not looking to build anything big, B# is the best language for simple IoT applications.

3. C

As you might have guessed, a lot of "things" won't really exist without one of the most important programming languages, C. It's basically a starting point and is the most popular language for embedded devices.

C has been used with IoT boards like Arduino and it is used most often even though other languages may rank a lot higher.

4. C++

C doesn't have the processing power of an object oriented pre-processor like C++. As a result, it's used as a pre-processor for C to enable it to run higher level languages. It's easy to make loads of mistakes with this language, but it's still a favorite among programmers.

In the most common Linux projects and embedded programming, it enables layers of objects, abstractions, and layers. It's ideal for developers looking to extend their programming code for IoT and embedded code.

Further, C++ helps you use other languages including C#, D, Java, and Python. But that's just the tip of the iceberg as it encourages the use of many more languages.

5. Go

Sharing many similarities with C, Go is an embedded language that was developed by Google. What's cool about Go is that it's stronger than C and allows devices to work together to send and receive data in many channels simultaneously.

But there is still a significant disadvantage here as there's a high possibility of data loss or errors if it's not managed properly during the coding phase. But as the language continues to evolve, things may change in the near future.

6. Java

When it comes to coding, Java is probably the most popular language out there. So it's no surprise that it's a popular choice among IoT developers.

This is especially true when it comes to consumer IoT (as you can "write it once, and run it anywhere"), but it also has the potential to really flourish in industrial IoT.

Java is also a language that has borrowed coding techniques from Mesa, C, C++, and many others. Further, it's well known to enable debugging code on a computer and then moving it to a chip via a Java Virtual Machine (JVM).

This means that the code can run in several different places where JVMs are common like smartphones and servers. But in this instance, it can also be run on the tiniest of machines.

7. JavaScript

All HTML and web browsers today use JavaScript as their programming language. Although it's taken bits and pieces from other languages (like Python and C), you can say that it's a scripting language that shares other language libraries like Java.

This goes a long way in making devices interoperable and its extensive use in present programming only helps make things easier. The popular offshoot in IoT development has been Node.js as much of the work is focused on hubs and servers to gather the data and store it (if they are small hubs or sensors, they're probably running on Node.js).

Two microcontrollers that run JavaScript from the beginning are Espruino and Tessel. JavaScript is omnipresent in web apps and websites and now web developers can easily move on to IoT development without learning a new language.

8. Parasail

If your IoT application needs a language that supports parallel processing, Parasail is a good option. However, it's important for developers to understand the difference between concurrent and parallel processors.

You can see a similar syntax in languages like C#, Java or Python, but if your IoT application requires parallel processing, Parasail is the best option.

9. PHP

When you think of PHP, you usually think of website prototypes and blogs, not IoT. But that's the reality as a lot of developers are now including a PHP code in their stack.

It's kind of an obvious selection as the code's main purpose is to juggle microservices on the server. Raspberry Pi developers are also now using LAMP on top of Linux turning something considered to be lowliest on the internet into a full blown web server.

Putting a LAMP stack on a chip also makes it easier to develop as the Raspberry Pi has enough spare cycles. Further, all server-side code that was developed over the last couple of decades can also be housed on a tiny sensor. Pretty amazing isn't it?

10. Python

A few years ago, no one thought Python would be used for IoT as it mainly focused on web applications. But that's changed now as it's essentially an easy programming language to understand and utilize in IoT projects.

Although Python started out as a scripting language to glue code together, it has grown to be one of the primary languages used by a lot of developers. As small devices have limited computational power and memory, developers had to get creative to make life easy, so they ended up choosing Python.

As a result, it has grown in importance within embedded devices space while enabling developers to create apps that are able to deliver comprehensible data mining results.

These days, most of the popular microcontrollers are also utilizing Python. For example, there are even small versions like the MicroPython board (only a few square inches) and software package.

If you want to develop something cool for Alexa, you better brush up on your Python programming skills.

11. Rust

Like Google's Go, Mozilla also developed an (open source) language, Rust. Often considered to be a great imitator of Go, Rust is able to do some things that are not possible with the former.

Rust is enabled to share information among different channels automatically. However, one drawback is that for Rust to function properly, the processor must be enabled to support concurrent processing.

12. Swift

Swift is the common language for developing iOS apps, so if you want it to interact via iPhones and iPads with your central home hub, Swift is the way forward. At the same time, its predecessor Objective-C will also work just as well.

As Swift gets more popular as a programming language for IoT, Apple also wants to be a leader when it comes to IoT at home. The company has been building infrastructure and libraries to handle much of the work, so it will make it easier for developers to just focus on the task and let the HomeKit platform handle the integration.

What’s the Best Programming Language for IoT Development?

The short answer is that it’s relative as all the languages above have their own role and influence within this space. But the ones that stand out will be the languages that support the end-use of the applications.

Right now Java is the programming language that’s leading the pack, but that can change over time. At the same time, it will also come down to the personal preference of the developer and the specific needs of the IoT project.