The Language of Codes : Why English is the Lingua Franca of Programming

As of August 2018, the total world population is 7.63 billion people and it continues to grow by the minute. About 1.5 billion people speak English worldwide, with approximately 360 million native speakers. Apart from being widely spoken, English is also the most commonly studied foreign language in the world.
According to the Ethnologue: Languages of the World, a catalog of world languages, there are nearly 7,000 known languages. And yet, English remains dominant as the de facto standard in the global economy, especially in international business.
In the world of computer programming, English seems to be the lingua franca for coding. Regardless of the original programming language, most keywords are still in English.
Evans Data Corporation, which regularly conducts in-depth surveys of the global developer population, estimated that there are 23 million software developers worldwide in 2018. By 2023, they are projected to be 28 million.
About 4.5 million of them are based in the U.S., with the largest number working in Silicon Valley. Around 70% of these developers were born outside of the US.
While the US currently has the largest population of software developers, India's developer population will overtake the US by 2023. The top nation for growth is in China, where it is projected to grow between 6-8% leading up to 2023.
Recent foreign investments in technology by US companies are also spreading beyond Silicon Valley to the fastest growing tech hubs in Europe like Berlin, Dublin and Paris. China's rapid growth in advanced technology and the increasing number of Chinese-speaking developers could potentially change the status quo.
One of our favorite design projects has been the Root Robot, an educational robot specifically designed to provide an interactive way for children to learn to code. With the rising importance of STEM education for global leadership, we can't help but wonder if English will continue to be the prevailing language used in software development.
We think so, and here's why.

WHERE IT ALL BEGAN

Charles Babbage, an English mathematician, philosopher, inventor and mechanical engineer, originated the concept of a digital programmable computer. Considered the "father of the computer", he invented the first mechanical computer in the early 19th century. This eventually led to more complex electronic designs, that of which formed all the essential ideas of modern computers we know of today.
Another key figure in the history of computer programming is Ada Lovelace, a contemporary of Babbage. She has often been called the world's first computer programmer. The daughter of the poet Lord Byron and his first wife, Anne Isabella Noel Byron, Lovelace was a brilliant mathematician, a woman ahead of her time in the 1840s. She was chiefly known for her work on Babbage's proposed mechanical general-purpose computer, the *Analytical Engine*. Lovelace was a visionary, the first amongst her peers to recognize the full potential of a "computing machine". She wrote the first algorithm intended to be carried out by such a machine and proposed to use the data input that programmed the machine to calculate Bernoulli numbers, which is now considered the first computer program. She had also predicted that machines like the *Analytical Engine* could be used to compose music, produce graphics, and be useful to science.
For these two reasons alone, there are people who might use this as an argument for coding to have a historically "English lineage". And they are not wrong. The collective work of both Babbage and Lovelace were profoundly influential on modern-day computing.
There is also another highly believed (though considered false by historians) theory that the commonality of English in coding started during World War II. The computer industry did grow out of technology developed by the Allies, mostly used by the Americans and the British.

It's also been said that most people learn English to use existing programming languages, so it continued on in the same vein when creating new ones. By attempting to use other languages for coding, it would cut off effective communication amongst programmers of whom the majority are already used to a standard i.e. coding in English. However, there is a growing movement and advocacy for developing non-English code.

MAKING THE CODE IN ENGLISH

Most new codes are actually developed by English-speaking individuals. But not all programming codes are in English. Although most keywords are written in English, comments, variable user written classes and methods are often in the programmer's own language.

Over a third of programming language were developed in English speaking countries. But some of the well-known, highly-used coding languages were developed in non-English speaking countries e.g. Switzerland (PASCAL), Denmark (PHP), Japan (Ruby), Brazil (Lua), and The Netherlands (Python).

The French developed the programming language LSE (Langage Symbolique d'Enseignement – Symbolic Language for Teaching) was based on French keywords. Due to the lack of French knowledge, that programming language eventually died out. The use of English for PASCAL was a reaction to the failure of LSE.

Outside of learning language for the sake of learning, there are hundreds of non-English programs out there. Wikipedia has a comprehensive list of Non-English based programming languages.

CODE SPEAK

While researching on this topic, we have also perused numerous developer's forums for the experts' perspective. It uncovered a wealth of opinions by programmers on why they think English prevails in coding. Here are some choice quotes:

"Having English keywords keep the programming abstracted from normal language for us…We do NOT want local language keywords for programming. Anyone who has made their own programming language at university and tried local keywords will agree with me on how confusing it is to read the code."

"English words are used more often, thanks to the technology industry. Part of the problem is that changing the key words isn't enough. Foreign grammars would make rote translated statements confusing. Word order can be very different from one language to another."

"In a global economy market that computer development has become, a common language seems to make the most sense."

"One of the difficulties in allowing the keyword to be switched on the fly to a different language is that keywords have to be 'reserved' such that they cannot be used as identifiers. Switching languages on the fly would cause a different set of reserved words to come into play and has the very strong potential to introduce conflicts."

"English has the advantage that it's spoken by many hundreds of millions of people. A language using non-English key words probably wouldn't have the demand needed to support it. The only exception seems to be Chinese. It's spoken by even more people than English is, and the huge difference between English and Chinese would make learning English more of a barrier for Chinese learning to program."

"English is the international language of business, people write programs primarily for business of some sort."

"English is the default language and the de-facto standard. If you want to reach a large audience, you pick a language that is spoken by many. Just like a medic learns Latin, we learn English."

"This is because programmers need to communicate about their code. That's easy if there's a standard, and the standard is easy – it's English."

"It is because English is the lingua franca for the polyglot world. It is a pretty simple matter to convert reserved word lists into any language you want, but nobody bothers because it creates

other problems on the platform. I speak 5 languages and have worked in many different countries. English is simply convenient and easy for most people."
"English language seems to be well-adapted for programming. In most native languages, finding concise equivalent of keywords can really be a burden. Some languages might measure up but some languages can't just stand the test."
"Most of the commonly used programming language are English-based, and most of them are that way to appeal to an international audience."
"For the code itself...all English. This makes sense when you realize that the majority of online resources are in English and the fact that sharing code with people of other cultures becomes easier too."
"For the most part, Chinese code in "English" (that is, they code in the exact same version of the programming languages used in the U.S., with English-based keywords and so on). However, there are some programming languages that cater to Chinese to varying degrees e.g. ChinesePython."
"Code is written in the standard programming languages, although comments (and perhaps variable names) could be written in Chinese. No use re-inventing the wheel (where "the wheel" is C++, Perl, Python, etc.) when you can just learn a few keywords."
Sure sounds like most programmers around the world do not have a problem with using English keywords to code. Why create a new way of coding if no one understands it? If it isn't broken, why fix it, right?