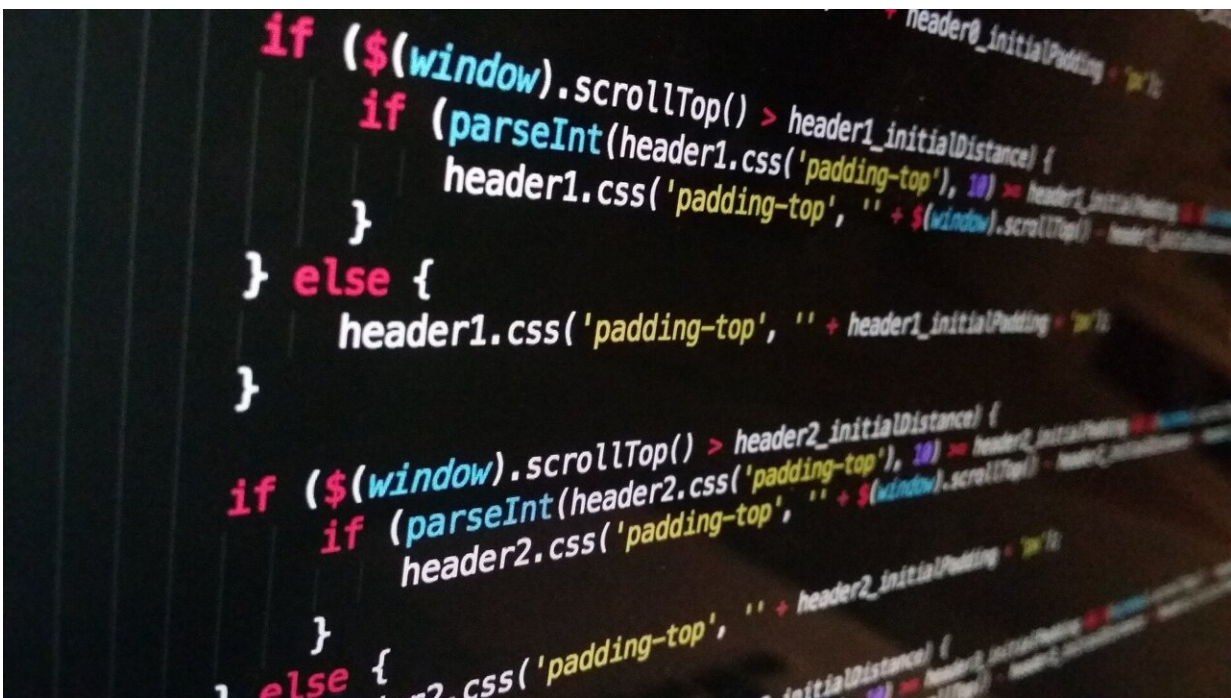


Q&A: Research shows neural connection between learning a second language and learning to code

April 23 2024, by Lauren Kirschman



Credit: Pixabay/CC0 Public Domain

As computer programming becomes an increasingly valued skill in the workforce, there is a greater need to understand how people learn to code most effectively.

Statistics show that up to 50% of students who enroll in introductory programming courses in the United States eventually drop out, suggesting a mismatch between how coding is learned and the way it's taught. A new study from the University of Washington, [published](#) March 5 in *Scientific Reports*, examines that issue.

The researchers recorded electrophysiological brain responses of varyingly skilled programmers as they read lines of code written in Python, a [programming language](#). The brain's response to viewing errors in both the syntax (form) and semantics (meaning) of code appeared identical to those that occur when fluent readers process sentences on a word-by-word basis, supporting a resemblance between how people learn computer and natural languages.

UW News spoke with co-authors Chantel Prat, a UW professor of psychology, and Chu-Hsuan (Iris) Kuo, a recent UW doctoral graduate of psychology, about their research, the future of teaching computer programming and more.

Why is it important to understand how learning computer programming works in the brain?

Iris Kuo: The idea of programming as literacy is something we wanted to focus on. We wanted to approach learning to program from a language learning perspective, specifically from a second language learning perspective. We've learned a lot about what makes a second language easy or difficult to learn and why some people are good at it and some people struggle. Now we're applying that lens to programming. If we can approach this topic from a different perspective, maybe we can address some myths or bring up new questions.

Chantel Prat: The idea of programming as the literacy of the future is

important. There's an increasing need and desire for programming in the workforce—as of 2016 over 20% of listed jobs required coding skills. It used to be this kind of niche skill that software engineers held, but now it's central to all STEM fields. Coding is a potential bottleneck to employment, but Intro to Programming continues to be one of these notoriously hard classes with high dropout rates. This is also a field where gender gaps are closing more slowly than other fields.

Everyone wants to tell you what it takes to be a good programmer, but many of their ideas aren't substantiated with science. Many of them are tied to culturally-linked ideas about who is already a good programmer. We know a lot about why and for whom learning a [natural language](#) is hard or why learning to read is hard. The question now was, can we leverage that expertise to start understanding how people with different levels of expertise understand code?

How did you conduct this research, and what were the main takeaways?

IK: There's a lot of literature in the second-language learning community that uses the event-related potential, or ERP, where we place sensors on people's heads and record their electrical activity to different stimulus. In this case, they were reading code. There are two distinct markers that indicate someone is processing meaning and when someone is processing form, like grammar. We wanted to use these two indicators to see if someone might react the same way while reading code.

If you're a native speaker of a language, or if you're really proficient, you tend to react to errors in meaning with a brain response marker called N400. You also tend to react to errors in grammar with a marker called P600. The more proficient you are in a language, the more distinct these markers are.

When you're first learning a language, you may be able to recognize something wrong with a sentence, but you may not be able to automatically process something as an error in meaning or grammar. Your brain takes time to learn these rules of grammar. Newer second-language learners tend to respond to most errors with the N400 marker, even when the error is grammatical. Over time, they learn to distinguish between something wrong with meaning and something wrong with grammar.

We wanted to see if something like that would happen with coding in people with a wide range of expertise. While all participants responded to errors in meaning and form in code, the higher their level of expertise, the stronger and more distinct their responses to the errors. This matches with what we have traditionally seen in second-language learners, where the more expertise you have in a natural language, the more sensitive you are to errors. This was the first study that realized we could have these neurological markers in coding and that people do process code incrementally.

CP: It was originally thought that N400 and P600 markers were language-specific. For a very long time, they were the gold standard for understanding brain processes associated with language comprehension. When research showed you can find them in certain cases for music and math, that was a huge deal. So, these markers aren't language-specific; they're about making meaning and how we understand what we take in incrementally.

Our study showed that when somebody reads a line of code with a bracket instead of a parenthesis, for example, their brain reacts in the same way as when they read a sentence with the wrong verb ending. And the fact that progression of sensitivity to form and meaning follows the same pattern as second language learning with increasing expertise is what we hoped to find, but it's still pretty exciting.

What does the future of this area of research look like, and what is the potential impact on coding education?

IK: We started with the coding language Python because it's one of the fastest-growing programming languages and one of the simpler languages for people to learn. It was designed to be really reader friendly. But the reality is, there are hundreds of other programming languages that serve different purposes. Some programming languages are more difficult or easier to learn, just like natural languages. We're working toward looking more extensively at the brain and seeing if our results can be replicated with other languages. I think this could impact the way we teach it.

Let's say a language is more reliant on structure, can you teach it the same way you teach something like Python? If we want to approach it from a language learning lens, how would we adapt that to accommodate something like Java, which is maybe more difficult for some people to learn?

CP: People have been talking about the gap between the way coding is taught and the way it's best learned since at least the 1980s. Coding education originated in an engineering culture—specifically a software engineering culture. Moving forward, there's good reason to support the idea of coding as learning a language, like learning to speak with computers. It should be taught like a language where you have elements of learning syntax, but you also have a lot of practice and "conversation" classes where you produce code in small groups. This also creates the option of using coding courses to fulfill second language requirements.

There may not be a one-size-fits-all best practice for [computer programming](#) education, but I think it's useful to understand the way

different people learn through a [second-language](#)-learning model.

More information: Chu-Hsuan Kuo et al, Computer programmers show distinct, expertise-dependent brain responses to violations in form and meaning when reading code, *Scientific Reports* (2024). [DOI: 10.1038/s41598-024-56090-6](#)

Provided by University of Washington

Citation: Q&A: Research shows neural connection between learning a second language and learning to code (2024, April 23) retrieved 29 June 2024 from <https://medicalxpress.com/news/2024-04-qa-neural-language-code.html>

This document is subject to copyright. Apart from any fair dealing for the purpose of private study or research, no part may be reproduced without the written permission. The content is provided for information purposes only.