

Lily Zhang
écrit le 21 mars 2016



When I took my first computer science class, I was struck by the similarities between learning programming languages and learning foreign languages. Our professor introduced Java statements much like my French teachers taught vocabulary words; the rigid rules of syntax and formatting were reminiscent of the many notebooks of grammar exercises I had filled in my study of the French language.

Both types of language revolve around communication. The purpose of learning French is to give a new exterior appearance to an idea through translation while keeping its interior meaning the same; the point of learning Java is to express the programmer's intentions, comprehensible to humans but not to machines, in a way that a computer can understand and execute.

Beyond semantics and syntax, there is great resemblance between the skills that are cultivated in foreign language and the skills that are utilized in technological fields. After earning my degree, I hope to pursue a career in software engineering, an industry that is based upon the ability to think in a simultaneously free and grounded way.

Learning foreign language is comprised of both creative and rigid elements highly applicable to software engineering. On one hand, the study of grammar teaches discipline and correct conduct. Good grammar is the linguistic version of the ability to abide by a set standard, parallel to the necessity of following protocol, industry standards, and the limitations of reality. Adhering to a set of rules gives us a mental scaffolding from which we train our minds to think rationally, logically, and orderly.

At the same time, composing phrases is a creative act. A student combines the vocabulary and grammar she has learned with her own personal sense of judgment to form a phrase, question or remark. It's an act that has the gravity of pushing back frontiers. And though her phrase may not be truly original, its formation is a unique instance, the result of a single moment's combination of impetus and circumstance. Likewise, software engineering is ultimately about designing solutions with constraints of context and resource in mind.

When I was still learning rudimentary vocabulary, my French teacher encouraged me to use words I did know in place of words I hadn't yet learned. Her advice taught me how to use my limited knowledge to express

sophisticated ideas rather than relying on the convenience of a dictionary. In the same way, it's not possible to master every quirk of every programming language. But with a little innovative spirit, you can adapt what you *do* know to solve a complicated problem or write an elegant solution. It's not a clever trick so much as the ability to adapt to a challenge using what you have, but one that is endlessly advantageous in any profession.

In my early French classes, I often had the burning desire to participate but never raised my hand because I was too timid. Building up my competence through years of study and practice gave me the confidence to freely express myself. In comparison to the journey of developing a correct French accent and internalizing countless words and grammatical exceptions, relaying my ideas has become child's play. The throaty *Rs* and musical intonations that terrified me as a beginner gradually became simple facts of life.

French is a subject that I have always been and always will be passionate about. In the half decade of courses I've taken so far, I've learned an immense amount. At the same time, my experience learning a foreign language has shaped me into a young woman with big dreams and the means to achieve them.