Home
Essays
H&P
Books
YC
Arc
Bel
Lisp
Spam
Responses
FAQs
RAQs
Quotes
RSS
Bio
Twitter

PAUL GRAHAM

Weird Languages

August 2021

When people say that in their experience all programming languages are basically equivalent, they're making a statement not about languages but about the kind of programming they've done.

99.5% of programming consists of gluing together calls to library functions. All popular languages are equally good at this. So one can easily spend one's whole career operating in the intersection of popular programming languages.

But the other .5% of programming is disproportionately interesting. If you want to learn what it consists of, the weirdness of weird languages is a good clue to follow.

Weird languages aren't weird by accident. Not the good ones, at least. The weirdness of the good ones usually implies the existence of some form of programming that's not just the usual gluing together of library calls.

A concrete example: Lisp macros. Lisp macros seem weird even to many Lisp programmers. They're not only not in the intersection of popular languages, but by their nature would be hard to implement properly in a language without turning it into a dialect of Lisp. And macros are definitely evidence of techniques that go beyond glue programming. For example, solving problems by first writing a language for problems of that type, and then writing your specific application in it. Nor is this all you can do with macros; it's just one region in a space of program-manipulating techniques that even now is far from fully explored.

So if you want to expand your concept of what programming can be, one way to do it is by learning weird languages. Pick a language that most programmers consider weird but whose median user is smart, and then focus on the differences between this language and the intersection of popular languages. What can you say in this language that would be impossibly inconvenient to say in others? In the process of learning how to say things you couldn't previously say, you'll probably be learning how to think things you couldn't previously think.

Thanks to Trevor Blackwell, Patrick Collison, Daniel Gackle, Amjad Masad, and Robert Morris for reading drafts of this.

- Japanese Translation