

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

# PAUL GRAHAM

## WHAT LANGUAGES FIX

Kevin Kelleher suggested an interesting way to compare programming languages: to describe each in terms of the problem it fixes. The surprising thing is how many, and how well, languages can be described this way.

**Algol:** Assembly language is too low-level.

**Pascal:** Algol doesn't have enough data types.

**Modula:** Pascal is too wimpy for systems programming.

**Simula:** Algol isn't good enough at simulations.

**Smalltalk:** Not everything in Simula is an object.

**Fortran:** Assembly language is too low-level.

**Cobol:** Fortran is scary.

**PL/1:** Fortran doesn't have enough data types.

**Ada:** Every existing language is missing something.

**Basic:** Fortran is scary.

**APL:** Fortran isn't good enough at manipulating arrays.

**J:** APL requires its own character set.

**C:** Assembly language is too low-level.

**C++:** C is too low-level.

**Java:** C++ is a kludge. And Microsoft is going to crush us.

**C#:** Java is controlled by Sun.

**Lisp:** Turing Machines are an awkward way to describe computation.

**Scheme:** MacLisp is a kludge.

**T:** Scheme has no libraries.

**Common Lisp:** There are too many dialects of Lisp.

**Dylan:** Scheme has no libraries, and Lisp syntax is scary.

**Perl:** Shell scripts/awk/sed are not enough like programming languages.

**Python:** Perl is a kludge.

**Ruby:** Perl is a kludge, and Lisp syntax is scary.

**Prolog:** Programming is not enough like logic.

■ [Japanese Translation](#)

■ [French Translation](#)

■ [Portuguese Translation](#)

---