

To some of us, writing computer programs is a fascinating game. A program is a building of thought. It is costless to build, it is weightless, and it grows easily under our typing hands.

But without care, a program's size and complexity will grow out of control, confusing even the person who created it. Keeping programs under control is the main problem of programming. When a program works, it is beautiful. The art of programming is the skill of controlling complexity. The great program is subdued—made simple in its complexity.

Many programmers believe that this complexity is best managed by using only a small set of well-understood techniques in their programs. They have composed strict rules (“best practices”) prescribing the form programs should have, and the more zealous among them will consider those who go outside of this safe little zone to be *bad* programmers.

What hostility to the richness of programming—to try to reduce it to something straightforward and predictable, to place a taboo on all the weird and beautiful programs! The landscape of programming techniques is enormous, fascinating in its diversity, and still largely unexplored. It is certainly dangerous going, luring the inexperienced programmer into all kinds of confusion, but that only means you should proceed with caution and keep your wits about you. As you learn there will always be new challenges and new territory to explore. Programmers who refuse to keep exploring will stagnate, forget their joy, and get bored with their craft.

Why language matters