CHAPTER 31

Keep an Open Mind to Go with Those Open Classes

Throughout the writing of this book I have been haunted by a feeling of irony. On page after page I have been talking up the technical flexibility of Ruby. I've gone on and on about how wonderful it is to program without the straitjacket of static typing, about how being able to change your classes on the fly makes for the magic of metaprogramming, and about how the flexible syntax enables us to build really nice DSLs.

But on these same pages I have also been laying down the law: Indent your code this way, be careful with `method_missing`, and avoid those class variables. Do what I tell you to do or you will put an eye out.

Stand back a little and the irony goes away: Ruby is a very, very sharp programming tool, and it pays to know what you are doing before you pick it up. If you are a relative newcomer to the language you will do well to build on the work of those who have come before you. So we have books like this one, full of rules and guidelines. But the rules and guidelines are only a means to an end. Really mastering a tool is not simply about knowing the rules and always following them. Real mastery comes when you know the rules and follow them—except in those rare moments when it's time to throw the rule book away.

In that spirit, I'd like to leave you with one last bit of advice. It's from George Orwell, who closed an essay on writing good English prose with this:

> Break any of these rules sooner than say anything outright barbarous.[1]

[1] Orwell, G. *A Collection of Essays.* San Diego, CA: Mariner Books, 1970.

If you ever find yourself in a situation that makes you want to throw this rule book away, visit www.eloquentruby.com or write to me at `russ@russolsen.com`. I'd love to hear about it.