Quotes from *Language-oriented Programming in Racket*

A new language literally makes possible new thoughts and new ideas and thereby increases the mental power of the language users. These new powers will need time to be assimilated.

Martin Ward

That kind of gradual development from an 'almost identical to Racket' language to one that has all of the features that you need, and behaves differently (possibly very differently) from Racket—that's where most of the benefits of LOP come in, and that's where Racket's approach shines.

Eli Barzilay

I've always been fascinated with different programming languages and even with the idea of creating my own, but I always saw it, and much of the literature and the public discourse presented it, as some impossibly complicated task not meant for mere mortals.

Annaia Berry

Racket provides a parser generator library but what makes Racket different as a language building material is two features: (i) its macro system, and (ii) the possibility to override what function application means.

Jörgen Brandt

I really had no exposure to the idea of DSLs. Rather, I wanted a certain result, and crept up on DSLs accidentally—it just seemed like the most direct way to do what I was after.

Matthew Butterick

Certain vocabulary emerges out of rudimentary Racket functions. It is like the Racket language is enriched in the direction of game playing, population matching, regenarating, etc. They carry significant meaning outside the defined range of original Racket.

Nguyen Linh Chi

If the programming model affects the structure of program fragments significantly then a new library is a sub-optimal solution as it introduces layers of encodings. A new language can hide all this and result in concise representations.

Christos Dimoulas

I saw that I had an upcoming need to write lots of variations of the same kind of program, and wanted to make those programs as human-legible as possible, which means abstracting away overhead tasks and having logical names for things.

Joel Dueck

I want [my students] to see that there are 'languages' everywhere. Sports teams have languages for planning plays; recipes are languages; medical protocols are languages.

Kathi Fisler

After you use higher-order functions for a while, you're not willing to go back. Macros are like that, so I think it's just a matter of time for enough programmers to catch on.

Matthew Flatt

I find making languages to be the same as normal programming: sometimes is easy, sometimes its hard. And often when its hard its because you picked the wrong design at the start!

Spencer Florence

I was kind of doing LOP anyway when I create new libraries for people. When I discovered Racket and began researching it, I realized I could create more than just functions for people. I realized I could create new syntax and basically make the "shape" of the code be anything I wanted it to be.

Stephen Foster

'Language-oriented programming', to me, means tweaking, augmenting, or changing an underlying computational model—or coming up with a completely new computational model.

Tony Garnock-Jones

I understand the term ‘language-oriented programming’ as an approach to problem solving where, before actually solving the problem, one focuses on expressing this problem in an optimal way.

Panicz Godek

I’d already gone from ‘thinking about problems in terms of data structures’ to ‘thinking about problems in terms of types’ and so when I heard Racket was good for ‘thinking about problems in terms of a tailor-made language’ that just sounded good.

Ben Greenman

Having a common language meant the ops team was free to tweak the model and draft patterns while I bolted a CSV file parser onto a finite-state machine. Without explicit formalism, we solved a complicated problem elegantly by developing a pattern language together.

Eric Griffis

I think it's the case that anyone engaged in 'bottom up' programming is actively doing 'language-oriented programming.' As a result, you can look at virtually any library out there and see a language in it. The types, and functions exported are the primitives used to build up layers of abstraction to form more complicated ideas.

Andrew Gwozdziewycz

I was always interested in programming languages and liked to learn about new ones, but the idea of realistically making and using potentially many languages never seemed practical until I learned about embedded DSLs made with macros.

William Hatch

it is relatively seamless to go between the two: to take a 'library' and add some 'language' behavior to it, or to take a 'language-enriched library' and just drop the 'language parts'.

Shriram Krishnamurthi

My language design is almost entirely oriented around some sort of optimization that I want to do. I need to restrict the language so the optimization applies then I need to build a language that enforces that restriction.

Jay McCarthy

Think about the countless hours poured into writing transpilers for JavaScript. All that effort to sidestep interpreters when a DSL or macro system could have achieved the same. Give me a fully-featured and production ready RacketScript any day.

Darren Newton

I think there are two ways of language-oriented programming. There are the macros you write to make your own code easier, and there are the languages you write for others to use. I've used Racket for both.

Pavel Pancheka

I understood from a long way back that rather than simply 'coding' in a given target language there was virtue in bringing the descriptive language to the problem. This can begin with functional abstraction and a ruthless desire to eliminate duplication and boilerplate.

Daniel Prager

The correct justification [of 'homoiconicity'] is that parentheses facilitate careful metaprogramming, the 'compile-time' creation of code. And metaprogramming is at the heart of what makes Racket great.

Prabhakar Ragde

I don't think 'making a language' is a yes-or-no decision: You create abstractions, at some point you abstract over syntax and maybe later down the path you discover that just doing require is awkward or repetitive, which is when you do a language.

Michael Sperber

The interface was originally provided strictly as a library, but composing using it felt like shoehorning the patterns into a program: quoting, boilerplate, etc. So I built a `#lang` interface instead.

Vincent St-Amour

Originally, I thought language-oriented programming was just about using macros to build interesting dialects of Racket, such as Typed Racket or Lazy Racket. But it's not really about writing dialects necessarily, since the languages people make are often hidden as a tool used to implement some other program.

Asumu Takikawa

I find Racket's ecosystem a good proof-by-construction of the feasibility of such a thing, considering the impressive tower of mini-languages that are used in concert.

Éric Tanter

The most frequent sticking point is the question of when something is a language versus a library. I adopt a liberal definition of a language that blurs the distinction between the two, as I find that this makes the concept of making one's own language more approachable.

Emina Torlak

What's nice for me about the Racket approach is that I only have to deal with the details of those aspects of the language implementation that I want. If I don't want to deal with lexing and parsing, for example, I can use the reader.

Jesse Tov

The particular domain of language construction might be especially amenable to DSLs. (There's a reason yacc stands for '*yet another* compiler compiler.' And yacc is old.)

Jon Zeppieri

Want more? Get a copy of *Language-oriented Programming in Racket: A Cultural Anthropology* **here**!