

Published on *InfoWorld* (<http://www.infoworld.com>)

[Home](#) > [Application Development](#) > [Languages and Standards](#) > 7 programming languages on the rise > 7 programming languages on the rise

7 programming languages on the rise

By [Peter Wayner](#)

Created 2010-10-25 03:00AM

In the world of enterprise programming, the mainstream is broad and deep. Code is written predominantly in one of a few major languages. For some shops, this means [Java](#) [1]; for others, it's [C#](#) or [PHP](#) [2]. Sometimes, enterprise coders will dabble in [C++](#) or another common language used for high-performance tasks such as game programming, all of which turn around and speak SQL to the database.

Programmers looking for work in enterprise shops would be foolish not to learn the languages that underlie this paradigm, yet a surprising number of niche languages are fast beginning to thrive in the enterprise. Look beyond the mainstays, and you'll find several languages that are beginning to provide solutions to increasingly common problems, as well as old-guard niche languages that continue to occupy redoubts. All offer capabilities compelling enough to justify learning a new way to juggle brackets, braces, and other punctuation marks.

[[Keep up on key application development insights with the \[Fatal Exception blog\]\(#\) \[3\] and \[Developer World newsletter\]\(#\) \[4\]. | See how the latest \[Python IDEs\]\(#\) \[5\] and \[PHP tools\]\(#\) \[6\] fared in our recent \[InfoWorld Test Center reviews\]\(#\). \]](#)

While the following seven niche languages offer features that can't be found in the dominant languages, many rely on the dominant languages to exist. Some run on top of the Java Virtual Machine, essentially taking advantage of the Java team's engineering. And when Microsoft built [C#](#), it explicitly aimed to make the virtual machine open to other languages. That detail may help make deployment easier, but it doesn't matter much to the programmer at creation time.

Either way, these seven languages are quickly gaining converts in the enterprise. Perhaps it's time to start investigating their merits.

Programming languages on the rise: Python

There seems to be two sorts of people who love Python [7]: those who hate brackets, and scientists. The former helped create the language by building a version of Perl [8] that is easier to read and not as chock-full of opening and closing brackets as a C descendant. Fast-forward several years, and the solution was good enough to be the first language available on Google's AppEngine [9] -- a clear indication Python has the kind of structure that makes it easy to scale in the cloud, one of the biggest challenges for enterprise-grade computing.

[For a look at the wide-ranging flock of Python IDEs, see "InfoWorld review: Nine fine Python development tools [7]."]

Python's popularity in scientific labs is a bit hard to explain, given that, unlike Stephen Wolfram's Mathematica for mathematicians, the language never offered any data structures or elements explicitly tuned to meet the needs of scientists. Python creator Guido von Rossum believes Python caught on in the labs because "scientists often need to improvise when trying to interpret results, so they are drawn to dynamic languages which allow them to work very quickly and see results almost immediately."

Of course, a number of libraries that offer much of what a scientist could want are available for Python. NumPy and SciPy are just two of the most notable libraries nurtured as open source projects and tuned for scientific computation.

Scientific and engineering enterprises such as pharmaceutical companies aren't the only ones tapping Python for research. Many Wall Street firms now rely heavily on mathematical analysis and often hire university scientists who bring along their habit of coding in Python. Python is becoming so popular on Wall Street that there are even proposals to require the prospectus for a bond to include a Python algorithm for specifying who gets what return on the investment.

Programming languages on the rise: Ruby

Some may argue that Ruby and Python are hardly "niche" languages, but the truth is, from an enterprise perspective, they remain promising tools all too often pushed to the margin. That said, Ruby, or more precisely the combination of Ruby with the Rails framework known as Ruby on Rails, is becoming increasingly popular for prototyping. Its entrance into the enterprise came on the heels of the Web 2.0 explosion, wherein many websites began as experiments in Ruby. 37signals [10] -- one of Ruby's many proponents -- actually uses Ruby to deploy code.

[The InfoWorld Test Center puts nine Rails IDEs and editors through their paces in "Lab test: Climb aboard Ruby on Rails [11]."]

The secret to Ruby's success is its use of "convention over configuration," wherein naming a variable `foo` causes the corresponding column in the database to automatically be named `foo` as well. As such, Ruby on Rails is an excellent tool for prototyping, giving you only one reason to type `foo`. Ruby on Rails takes care of the rest of the CRUD scaffolding for you.

Ruby on Rails sites are devoted to cataloging data that can be stored in tables. Well-known examples include Web applications like Basecamp, Backcamp, and Campfire from 37Signals, a

collection of websites that knits together group discussions, debates, and schedules. Ruby on Rails handles the formatting of these database tables, as well as decisions about what information to display. Using Ruby on Rails' naming convention, production quality code can be sketched up easily without much duplicate effort.

Many of the production-grade Ruby websites run on JRuby, a version written in Java that sits squarely on the JVM. JRuby users get all of the JVM's prowess in juggling threads, a very valuable asset in production-level deployments with many concurrent users.

Programming languages on the rise: MATLAB

Built for mathematicians to solve systems of linear equations, MATLAB has found rising interest in the enterprise, thanks to the large volumes of data today's organizations need to analyze [12]. Many of the more sophisticated statistical techniques that match people with advertisements, songs, or Web pages depend upon the power of algorithms like those solved by MATLAB.

Expect MATLAB use to grow as log files grow fatter. It's one thing for a human to look at the list of top pages viewed, but it takes a statistical powerhouse to squeeze ideas from a complex set of paths. Are people more likely to shop for clothes on Monday or Friday? Is there any correlation between product failures and the line that produced them?

MathWorks, the company behind MATLAB, offers a diverse set of whitepapers showing how engineers are searching for statistical answers. Toyota Racing [13], for instance, plans its NASCAR entries by analyzing tests in wind tunnels and other labs. Canada's Institute for Biodiagnostics [14] is searching for the best treatment for burns.

There are also a number of open source alternatives, including Octave, Scilab, Sage, and PySci, one of the aforementioned Python libraries [15]. All of these tools help with the complicated statistical analysis that is now becoming common for firms trying to understand what the customer did and what the customer may want to do in the future.

Programming languages on the rise: JavaScript

JavaScript is not an obscure language by any means. If anything, it may be the most compiled language on Earth, if only because every browser downloads the code and recompiles it every time someone loads a Web page. Despite this fact and the increasing dominance of AJAX-savvy Web pages, JavaScript is rarely thought of as a language that runs on the big iron.

This isn't for lack of trying. Netscape tried to make JavaScript the common language on its server platform back in 1996, but ended up establishing it only in the browser. Aptana, one of the latest devotees, throttled its development of Jaxer when it never caught on. AppJet, a small experimental company, used the Rhino JavaScript library written in Java to make it simpler to code server-side. That company was acquired by Google [16] in 2009 and now seems to be devoted to other projects.

Still, new applications for JavaScript abound. CouchDB [17], for instance, doesn't use SQL for queries, instead taking two JavaScript functions, one for selection (Map) and the other for

bundling everything together (Reduce). [Node.js](#) [18] is one of the more exciting server-side JavaScript frameworks [19] to appear as of late, revitalizing the ancient dream of bringing harmony to both client and server-side programming. The package takes Google's V8 JavaScript engine created for the browser and lets it make the decisions about formatting outgoing data.

Everywhere people need a small amount of scripting power, JavaScript finds new uses. One of the simplest ways for developers of large applications to offer users the ability to create subapplications, JavaScript continues to grow in the enterprise, one small chunk of code at a time.

Programming languages on the rise: R

Statistical analysis is being increasingly done in R these days, although some purists call the language S, its original name. Tibco sells a commercial version called [S-Plus](#) [20].

[For an in-depth primer on what enterprises are doing with "big data," see "[The big promise of Big Data: What you need to know today](#) [12]."]

There probably won't be an S++ because the language is more a version of LISP or Scheme with additional features for computing statistical functions and then displaying the results in pretty pictures. If the boss wants the computer to churn through billions of lines of log files looking for patterns, clusters, and predictive variables, R or S is a well-loved solution.

R is another Swiss Army Knife of numerical and statistical routines for hacking through the [big data sets](#) [12] -- collections big enough that it might be better called a Swiss Army Machete. Lou Bajuk-Yorgan, senior director of product management for Tibco's Spotfire S-Plus, says its software is used by a number of clients who are studying how business or engineering projects might work or why they fail to work. Analyzing weather patterns to find the best places to build wind-powered generators is one example.

Programming languages on the rise: Erlang

Does your server need to respond to many different independent messages concurrently? Do you need to parcel these requests out to different cores or servers in various parts of the world? That's practically the definition of the hardest part of enterprise computing. [Erlang](#) [21], an open source language first created by scientists at Ericsson Computing Laboratory, excels at these tasks.

[For a deeper look at today's "slacker databases," see the [InfoWorld Test Center review "NoSQL databases break all the old rules](#) [22]."]

The language mixes traditional facets of functional programming (no side effects) with a modern virtual machine that compiles down to machine code. The structure of the language forces the programmer to build something that's easier to spread across multiple cores and multiple machines. There are a number of practical implementations of [Web servers](#) [23] and the [CouchDB](#) [24]. That's right: The database that asks to receive queries written in JavaScript instead of SQL is itself written in Erlang.

CouchDB is just the beginning. A number of project managers dealing with "big data" are building systems for storing large volumes of data in a scalable way. [Hibari](#) [25], an open source project from [Gemini Mobile](#) [26], offers consistent, scalable clusters to store key-value pairs that repair themselves after failure. The functional structure makes it easier to create big applications that juggle multiple connections efficiently.

Programming languages on the rise: Cobol

It may not be fair to call Cobol a niche language as it was once the dominant language in the enterprise. Grace Murray Hopper, famous for finding the first bug in the early mainframes, helped create the language in 1959 and it's been enhanced hundreds of times since. Cobol jockeys today get to play with object-oriented extensions, self-modifying code, and practically every other gimmick.

That never earned it much respect in some circles. Or as famous academic Edsger Dijkstra put it: "The use of Cobol cripples the mind; its teaching should, therefore, be regarded as a criminal offense." The folks in mainframe shops everywhere ignored this note and soldiered on. IBM calls one of the latest releases "Enterprise Cobol 4.2," but it could as easily be numbered 147.2 or maybe even 588.3. Cobol programmers like the syntax that's more like a natural language with actual nouns and verbs that form clauses and sentences -- a technique that might call Ruby to mind.

While fewer schools are teaching new programmers Cobol, the language is [far from dying](#) [27], with many corporations continuing to invest in their Cobol stacks. A recent search of Dice.com showed 580 jobs mentioning Cobol and 1,070 mentioning Ruby. The bulk of the jobs seemed to involve counting money ("asset management") and counting doctor's visits ("Health IT"). While these are some of the same areas that first adopted computers for back-office processing, the work still needs to be done.

Versions of the languages run on JVMs and .Net virtual machines making it possible to [migrate code stacks away from mainframes to Linux boxes](#) [28]. Programmers who want to use a more modern IDE can search for [plug-ins to Eclipse](#) [29], a project that is gaining new support.

Programming languages on the rise: CUDA extensions

As libraries for programming video cards to do massively parallel jobs, CUDA extensions are not technically a language; they're just extensions to C. Still, some enterprise programmers are beginning to unlock the massively parallel architectures normally devoted to rendering realistic blood splattering in alternative game worlds. Moreover, recoding loops for massive parallelism means rethinking many of the idioms from basic C or C++ programming, making CUDA extensions all the more valuable.

Opportunities to tape CUDA extensions include machine vision, massive simulations, and huge statistical computations. Many problems of data analysis are naturally massively parallel, making GPU processors worth a look. One of Nvidia's recent conferences devoted to CUDA applications included separate tracks devoted to computational fluid dynamics, computer vision, databases and data mining, finance, and molecular dynamics. That list alone is long

enough to explain why big enterprise coders are curious.

"It's clear that the GPU reached escape velocity," Dan Vivoli, senior vice president at Nvidia, said at the Nvidia conference after scientists had presented papers on how the GPU's parallelism can work in these domains. "The processor is now reaching all different disciplines of science and industry."

Related articles

- [InfoWorld review: Eight PHP power tools](#) [30]
- [InfoWorld review: Nine fine Python development tools](#) [31]
- [Lab test: Climb aboard Ruby on Rails](#) [32]
- [The big promise of Big Data: What you need to know today](#) [33]
- [NoSQL databases break all the old rules](#) [34]
- [InfoWorld review: Databases primed for social networks](#) [35]

This article, "[7 programming languages on the rise](#) [36]," originally appeared at [InfoWorld.com](#) [37]. Follow the latest news in [programming](#) [38] at [InfoWorld.com](#).

[Developer World](#) [Cobol](#) [Python](#) [Ruby](#) [Languages and Standards](#)

Source URL (retrieved on 2010-10-29 04:11PM): <http://www.infoworld.com/d/developer-world/7-programming-languages-the-rise-620>

Links:

[1] <http://www.infoworld.com/d/developer-world/infoworld-review-top-java-programming-tools-191>

[2] <http://www.infoworld.com/d/developer-world/infoworld-review-eight-php-power-tools-737>

[3] <http://www.infoworld.com/d/developer-world/blogs?source=fssr>

[4] http://www.infoworld.com/newsletters/subscribe?showlist=infoworld_developer&source=ifwelg_fssr

[5] <http://www.infoworld.com/d/developer-world/infoworld-review-nine-fine-python-development-tools-374?source=fssr>

[6] <http://www.infoworld.com/d/developer-world/infoworld-review-eight-php-power-tools-737?source=fssr>

[7] <http://www.infoworld.com/d/developer-world/infoworld-review-nine-fine-python-development-tools-374>

[8] <http://www.infoworld.com/t/languages-and-standards/whatever-happened-perl-012>

[9] <http://www.infoworld.com/d/cloud-computing/first-look-googles-high-flying-cloud-python-code-190>

[10] <http://37signals.com/>

[11] <http://www.infoworld.com/d/developer-world/lab-test-climb-aboard-ruby-rails-851>

[12] <http://www.infoworld.com/d/data-explosion/the-big-promise-big-data-what-you-need-know-today-585>

[13] http://www.mathworks.com/company/user_stories/userstory45540.html

- [14] http://www.mathworks.com/company/user_stories/userstory45502.html
- [15] <http://www.infoworld.com/d/developer-world/7-programming-languages-the-rise-620#python>
- [16] <http://www.infoworld.com/t/collaboration/google-buys-collaboration-app-boost-wave-129>
- [17] <http://www.infoworld.com/d/data-management/couchdb-emerging-top-choice-offline-web-apps-777>
- [18] <http://nodejs.org/>
- [19] <http://www.infoworld.com/d/developer-world/watch-out-java-here-comes-javascript-799>
- [20] <http://spotfire.tibco.com/products/s-plus/statistical-analysis-software.aspx>
- [21] <http://www.erlang.org/>
- [22] <http://www.infoworld.com/d/data-management/slacker-databases-break-all-the-old-rules-599>
- [23] <http://erlyweb.org/>
- [24] <http://couchdb.apache.org/>
- [25] <http://hibari.sourceforge.net/>
- [26] <http://www.geminimobile.com/>
- [27] <http://www.infoworld.com/d/adventures-in-it/7-dirtiest-jobs-in-it-937?page=0,1>
- [28] http://findarticles.com/p/articles/mi_pwwi/is_20050229/ai_mark01049402/
- [29] <http://www.eclipse.org/tools/proposals/cobol-proposal.html>
- [30] <http://www.infoworld.com/d/developer-world/infoworld-review-eight-php-power-tools-737?source=rs>
- [31] <http://www.infoworld.com/d/developer-world/infoworld-review-nine-fine-python-development-tools-374?source=rs>
- [32] <http://www.infoworld.com/d/developer-world/lab-test-climb-aboard-ruby-rails-851?source=rs>
- [33] <http://www.infoworld.com/d/data-explosion/the-big-promise-big-data-what-you-need-know-today-585?source=rs>
- [34] <http://www.infoworld.com/d/data-management/slacker-databases-break-all-the-old-rules-599?source=rs>
- [35] <http://www.infoworld.com/d/data-management/infoworld-review-databases-primed-social-networks-828?source=rs>
- [36] <http://www.infoworld.com/d/developer-world/7-programming-languages-the-rise-620?source=footer>
- [37] <http://www.infoworld.com/?source=footer>
- [38] <http://www.infoworld.com/d/developer-world?source=footer>