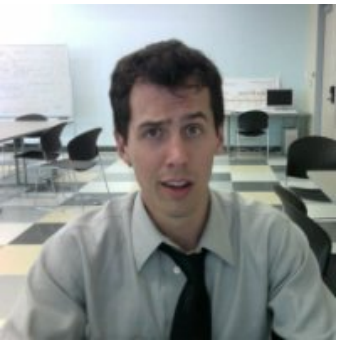
Scala enthusiasts

9,467 members | Member



What has Scala done for you or your organization?

David Kaczynski
Software Developer and IT Enthusiast

Did Scala help reduce the size or complexity of your code base? Was there something that Scala offered that an imperative language was unable to? Were you attracted to the libraries of asynchronous and concurrent support, like scalaz or akka? Did functional programming constructs help you represent complex states in mathematics? In a nutshell: why Scala?

+ Follow David

Like (4) • Comment (6) • Share • Unfollow • Reply Privately • 5 months ago

Comments

Michael Franz, Serge Bureau and 2 others like this

6 comments

David

David Kaczynski
Software Developer and IT Enthusiast

Personally, I picked it up as a hobby last summer after reading about it on HammerPrinciple.com: http://hammerprinciple.com/therighttool/items/scala

Since then, I've used it on pet projects during grad school. Coming from Java, my Scala code is much more compact, and I can do more with fewer lines of code. My work has mostly been in web services, and learning Akka has taught me a concurrency model that doesn't rely on locks and synchronized blocks.

Like (3) • Reply privately • Flag as inappropriate • 5 months ago

Dmitry S., Robin G. and 1 other like this

Satyarth

Satyarth Negi
Entrepreneur, Architect

I had the same question in my mind when i started to learn Scala (I am still very much a newbie). Here are the benefits of functional programming/Scala that i have come to realize :

-- Enables you to write logic in fewer lines.
-- As opposed to routine in earlier times, data is at the heart of softwares being written in modern times. So a programming approach which allows us to reverse the role i.e pass routine to data instead of data to routine is closer to the problem we are trying to solve.
-- Writing reusable code in OOP comes as a result of experience and multiple iterations that you do on your code over time. FP embraces code reusability as fundamental building block.
-- Scala embraces immutability of state/object in its design. So makes concurrent programming easier
-- Functional programming paradigm is better suited for writing parallel/distributed software.
-- Scala is the language of choice for someone coming from Java/OOP background to venture into functional programming

Like • Reply privately • Flag as inappropriate • 5 months ago

Alexei

Alexei Naybich
Software Developer, System Architect, Analyst

I started to take interest in Scala about two years ago. I was amazed reading the Odersky's bible "Programming in Scala, 2nd edition" and it soon became my hobby. For fun, I used to decompile some Scala constructs to see how it is desugared to low-level bytecode and found that the result

Top Influencers in this Group

Erik Peterson
MBA, BSEE, Executive, Entrepreneur, Architect, Engineer
Follow Erik

See all members

Picture Yourself at Comcast

COMCAST

Marcin Milewski
Engineering at Comcast

View Jobs | + Follow Company

Latest Activity

Nikolas Papirniy and 1 more commented on:
Review Copies available for newly published book on SBT for Scala by Packt
1 hour ago

Mark Lewis and 1 other like this comment by Ron Pomeroy Scala's rank in Tiobe index?: Anecdotal point: The last Seattle Scala Meetup went from roughly 14 to 40 participants. I'm not convinced the Tiobe Index is accurately...
2 hours ago

Slim Ouertani likes this discussion by Slim Ouertani Démystifier les iteratees avec Java
4 hours ago

See all activity

Subgroups

BeScala
68 members

Boston Area Scala Enthusiasts
103 members

See more

is extremely efficient.

Since then I've been trying to push my company's management to start using Scala in parallel with Java, wherever it is appropriate, but there were no proper new projects to try it at until recent time that I finally got a "green light" for a small sub-project to be built on top of our Java sub-systems.

Though being aware of Scala-only frameworks, I was focused to adapt the existing eco-system to which our Java department was used to, to Scala. Our team is rather agnostic to standard JEE technologies in favor of an alternative stack based on Spring, Hibernate (non-JPA), FreeMarker, etc. We also use IntelliJ IDEA and Maven in the development process.

The marathon took about two months and finished quite successfully. We succeeded to apply idiomatic Scala almost everywhere, with some exceptions like Hibernate collections which are still in Java. However I'm going to try introducing transparent Scala wrappers in addition, and then the solution might be considered as fully idiomatic. Though integration of Scala objects and collections with FreeMarker succeeded, we decided to try also another template engine Scalate with Jade syntax and got very positive experience resulted in that we implemented the presentation layer fully on Scalate and additionally started to create our own web component framework based on it.

Leaving out technical details, now I can say about my personal feelings. I've been into programming for about 25 years and I cannot remember any other technology that drove me to the "wow-effect" so much than experience with Scala did. The code became so clean and concise that I can't stop wondering. I suppose that my efficiency as a programmer increased at least twice because I don't waste time on boilerplating and making stupid mistakes which are so common in Java and almost impossible in Scala if one adheres to its idiomatic approaches.

So, everybody go in for Scala. It's not that complex as it may seem to newcomers.

Like (5) • Reply privately • Flag as inappropriate • 5 months ago

Michael Schuenck D., Călin-Andrei B. and 3 others like this

Stefano

Stefano Semeria
Contractor at Reply

I have no real experience with Scala. I started learning it as an hobby during spare time while using Java at work. I also used Groovy but only for simple scripts.
Learning Groovy made me aware of Java limits, so I became curious about Scala.
I read Odersky's book and I tried to follow his course on Coursera.

What I personally found great while exploring Scala is that removing the difference between methods and operators and invoking methods without the dot notation lets you literally "extend the language", creating new keywords and operators at will. This, together with a rock solid type system, enables writing code in a way that resembles a real domain specific language more than traditional code.

So, from a business point of view, the main benefits of writing Scala code are:
1) Shorter code (shorter development cycle)
2) Higher Level code (code is easier to understand, once you have learned the new syntax => less maintenance effort)
3) More Functional Code (less code side effects => more testable and debuggable code => less maintenance effort)
4) Scala makes you think in new ways => better programmers

I also found problems in adopting Scala on enterprise development teams:
1) Syntax is different from Java, so you have to spend time for training.
2) Paradigm shift. Learning a new language can be easy but learning a new way of writing programs is far more difficult. So an average team will probably feel only that they write less code after adopting Scala, but the other benefits will follow after a while. This makes hard producing good estimates about which is the real speed factor of the language on a traditional work cycle.
3) Psycological Barrier: Why an experienced Java programmer, should learn a new language with a funny syntax, in particular one that compiles to JVM bytecode?

In my opinion, the best way of introducing Scala at work is as a scripting language.
If developers are encouraged in writing small everyday scripts using Scala instead of, let's say Python, they can start the paradigm shift earlier and break the "psycological barrier" earlier.

So, thumbs up for enterprise Scala!

Unlike • Reply privately • Flag as inappropriate • 5 months ago

Manish L., Marcin Milewski like this

Pandu

Pandu Pradhana
Software Architect at Digitized

Our company invested in Scala as our primary language for our core engines. At first, we have experienced Java programmers in team with little experience in functional programming. But in the end, it gives a great way to solve problems. Writing in Scala produces fewer lines, thus making it easier to read, making it easier to maintain. One other thing that make Scala great is its concurrency frameworks: the Actors, Futures and Promises. For this, we use Akka.

Like • Reply privately • Flag as inappropriate • 5 months ago

Manish

Manish Lohar
Founder & CEO at VulcanMinds

Good points above! They however apply to techies looking to move on to Scala. The real answers David might be looking forward to unearth are the TCO and ROI related. How do you convince the senior folks to invest in Scala, Scala ecosystem, resources availability, development/testing processes, expertise availability, issues resolution etc etc. The hard numbers and comparative scenarios may however be elusive for the discerning though.

Like (1) • Reply privately • Flag as inappropriate • 5 months ago

Nicolas Marchildon likes this

Add a comment...

Send me an email for each new comment.

Add Comment