

Benefits of Scala



When a new programming language is created, it came from a desire of one person or a group of to do it better than their forerunner. It's an evolutionary process that humans are trying to improve themselves if something doesn't as they want it to. Sometimes it's only a small piece of improvement which creates an absolutely new feeling about something. By designing a new programming language you have a great pool of existing languages: LISP (1958), Smalltalk (1970), C++ (1979), Python (1991) and Ruby (1995). Today it is often told not to waste your time on building up an new language from the scratch. It's time-consuming and why should you invent the wheel a second time? I think this is just normal. If you spend some time reading about the languages mentioned above, you will see that older language have still a great influence on new ones. Today, it is common to use DSLs (they are used extensively in Rails) to create new languages. Not everything happened in the past was bad and sometimes if you stick your head into some old forgotten technology you may find some diamonds.

This was the main idea of Odersky when he created Scala. It is the first language which is a real hybrid language and combines the invigorations of OOP and FP.

Main facts about Scala

- Scala is ideal for today's scalable, distributed, component-based applications that support concurrency and distribution.
- Scala is *statically typed* that means that the type of some variable is immutable during the whole execution of the program.
- There is modular *mixin-composition for classes* - some hack to enable multiple inheritance in Scala, which solves the diamond problem through linearity of the inheritance hierarchy via *traits*.
- Liquid error: undefined method `join' for #
- Lower risk to use Scala in an existing Java Application because Scala works *seamless* with existing Java Code.
- high level type system with *variance annotations, compound types, lower and upper bounds for types*

- usage of *inner and anonymous classes*
- *implicit conversions* - that means a function take one type as an argument and returns to another type (like converting an Integer into String)

Conclusion

Scala is a very rich language which combines many features of different languages. It's great if you have to make different sections of your code cleaner, faster and conciser. See [Scala at LinkedIn \(http://www.scala-lang.org/node/6436\)](http://www.scala-lang.org/node/6436) to know what I mean.

Thanks to *Hanna Schuett* and [Sebastian Gerth \(http://sebastian-gerth.eu\)](http://sebastian-gerth.eu) for reading an earlier draft of this post.

Further reading

- [LISP \(http://en.wikipedia.org/wiki/Lisp_%28programming_language%29\)](http://en.wikipedia.org/wiki/Lisp_%28programming_language%29)
- [Smalltalk \(http://en.wikipedia.org/wiki/Smalltalk\)](http://en.wikipedia.org/wiki/Smalltalk)
- [C++ \(http://www.cplusplus.com\)](http://www.cplusplus.com)
- [Python \(http://www.python.org/\)](http://www.python.org/)
- [Ruby \(http://www.ruby-lang.org/en/\)](http://www.ruby-lang.org/en/)
- [DSLs \(http://en.wikipedia.org/wiki/Domain-specific_language\)](http://en.wikipedia.org/wiki/Domain-specific_language)
- [diamond problem \(http://en.wikipedia.org/wiki/Diamond_problem\)](http://en.wikipedia.org/wiki/Diamond_problem)