the best programming language

When I want to take a break at work, I sometimes read technology forums. And there is one kind of posts that I really like: the flame wars between programming languages. I like these posts because you can see passionate and smart people who are arguing as if their lives were at play.

These posts have 2 advantages:

- they make me laugh
- I learn new stuff

If I had to sum up this kind of posts, it would be something like:

Post Title “Java is the best language” by NewJavaFanBoy

> *NewJavaFanBoy: Java is the best language because of its community. Moreover, it has really cool features like lambdas. Why so many people hate Java?*
>
> *FormerJavaFanBoy: Oracle killed Java.*
>
> *DotNetFanBoy: The evolution of Java is too slow, C# had lambdas a while ago. Moreover, some critical features like optional and named parameters are not in Java. Now that dotnet is more open sourced and can be run on Linux with Mono, Java is going to die.*
>
> *TrollRoxxoR: BecauseJavaDevelopersDontKnowHowToWriteCode*
>
> *RealG33k: Both your languages are for kids, C++ is way better but it’s for real developers only. Do you even know what SOLID means?*
>
> *HipsterGeek: So old and lame … you should try Node.js, it’s based on asynchronous calls and it’s very fast.*
>
> *LinusTorvalds: Pussies, a real developer uses C or assembly. You can’t have performances with those high level shits.*

I hate PHP. I can't explain why; it must be because I tried to learn it when I was 14 and it messed with my brain. But guess what, you're reading this post on a server using PHP/NGINX (which is a kickass server by the way). I'm good with Java. So, I could have used a Java framework running on a ~~fast~~ fat JVM. But, WordPress is a great platform. It's often looked down by purists but it clearly answers my needs. The aim of my blog is not to be the fastest in the world (though it has surprisingly but painfully survived 2 Hacker News and Reddit front pages involving 500 simultaneous connections). I just want a user-friendly interface where I can share my thoughts.

Which leads to my point: there is no best programming language, it depends on the situation.

1) Do you need performances?

If yes, what kind of performances are we talking about?

- Seconds? Every language can do it!
- Milliseconds? Every language with good programmers can do it.
- Microseconds? At this step, you can remove all the interpreted languages (like python, which is a good language). I know that a well-tuned JVM with very good Java programmers can do it. I imagine that it's the same for C#. Of course, a pure-compiled language can deal with that.

But in all these cases the programmer's skills are more important than the language.

- Nanoseconds? Only assembly or maybe C can deal with that.

So, in most situations developers' skills are what matters.

2) What's about the ecosystem?

More than the language itself, the ecosystem is important.

I've used Visual Studio during my scholarship and I have been amazed by the coherency of Microsoft's ecosystem.

Now, I'm more an Eclipse guy. Even in the Java community, Eclipse is looked down by purists who now use IntelliJ IDEA. Eclipse is an open source software developed by different people and it's clearly visible (in a bad way). Compared to the coherency of Visual Studio, you'll find different logics in the different plugins of Eclipse.

But, if having tools is great, knowing how to use them is better. For example, when I started in Java, I was very slow. I learned by hearth some Eclipse keywords and it's changed my developer life. I've also looked for useful plugins, and Eclipse has plenty of them, because it's a rich ecosystem.

3) What's about the online help?

Ok, you're using your kickass programming language but don't tell me you know every side of this language by hearth. Having a well known language is useful when you need help. A simple Google or StackOverflow search and you get your answer by Ninja_Guru_666 and I_AM_THE_EXPERT. If you're more like an in-depth programmer, you can also check for the official documentation assuming it exists for the problem you're looking for.

4) What are the skills of the team?

If the developers don't really know how a computer works, using a compiled language is a suicidal move. And, compared to the purists, I don't see why knowing (exactly) how a computer works makes you a good developer (though, I must admit, it helps; but there are more important skills).

It's better not using the best tool but a known one. Moreover, many developers are fan boys. Using their preferred language will help them to stay motivated on the project.

5) The business side

An objective point of view is to see what the most in-demand languages are. It doesn’t mean they’re the best but at least you’ll get a job. In this case, Java, C#, PHP, SQL and JavaScript are clearly above all (at least in France).

Moreover, as a technical leader, it’s always good to check the skills in the market before choosing a technology. If you choose the best but rare technology to deal with your problem, good luck for finding skilled developers on the technology.

But what’s true in 2015 might change in 2018. ActionScript was a must have not so long ago. Likewise, with Swift, all the hours spent on Objective C will become obsolete in a few years.

To conclude, I’ll end up with a lame and (I hope) obvious conclusion: there are no best programming languages or best frameworks; what’s best now might not exist tomorrow. A programming language is just a tool; what matters is the way you overcome your problems.