What programming languages today?

Thomas GIGOUT
thomas.gigout@etudiant.univ-reims.fr

Wilfried DAUNIQUE
wilfried.daunique@etudiant.univ-reims.

Summary: *Nowadays, the programming languages studied in schools are quite limited in number. However, those we know today are the result of a great evolution of languages – like our languages that have evolved – from Latin to French for example.*

Key words : programming language; computer science; IT; coding.

1. Introduction

In computer science, and more specifically in computer programming, the choice of the language used is as important as the choice of the language you will use in a country. However, according to [1], there are more than 700 programming languages listed on [2]. The most popular index today for listing programming languages is the [3] index. That's why it's quiet legitimate to ask the question of the most popular, the most used and the most successful programming languages of our day.

But when you ask Google this question, the only things you can find are:

- Articles that list the x new popular languages of these last years like on [4] "5 emerging programming languages with a bright future".

- Articles that give a ranking of the most popular languages like on [5] "*The 5 Most Popular Programming Languages of 2019*".

These types of articles respectively answer only part of the question. Now, we can ask why those languages are so popular, why programmers choose one language over another, why a language is preferred and how to choose between all those languages. And this is what we are going to answer in this article. At first, we will try to provide some answers to "why a language is popular and preferred". Then, we will be able to analyse the "old" languages at the top of the rankings to finish with the emerging ones. But before all that, we present a state of the art on the subject. The aim of this study is to perform an analysis of programming languages today.

2. State of the art

As it was said in the introduction, [3] list all the popularity and ratings per language according to a lot of different parameters, like the number of searches on the internet, number of project identified using it (*source:* *https://www/tiobe.com/tiobe-index*). On this website, we can see all the ratings of different languages across the time. We can notice that the top is composed of those who are mostly created in the 90s, like Java and C. But even if we know they meet the index criteria to be in the top, we do not know why they fill them, why they are popular and so successful.

In the same way, we have [5] which, every quarter, offers a graph of the popularity of the most used languages according to [6] and [7].

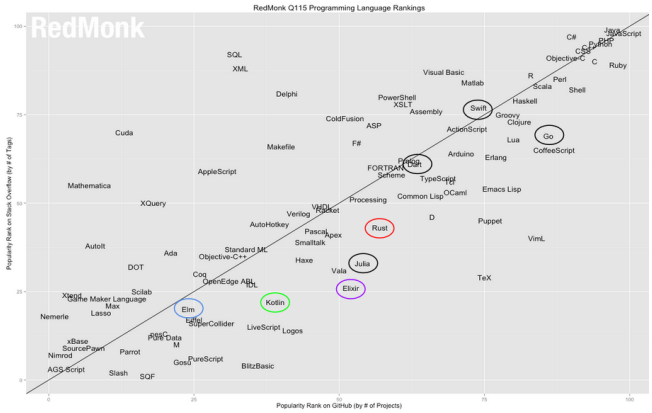

Figure 1 : ranking for 2015, Januray

Some languages are more popular on [6] and the others are more popular on [7]. But it's evolve permanently since there are updates each quarter. Important thing, we can see on this chart absolutely all the most popular languages, including the new ones that are gaining popularity.

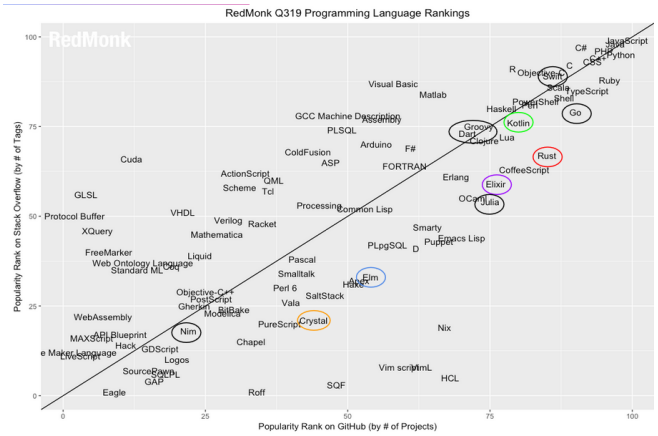

Figure 2: ranking for 2019, June.

With figures 1 & 2, we can notice that the top of the languages are still the same, JavaScript, C, Java etc… but we also see that some languages like Elixir or Kotlin exploded in 4 years and that they are now alongside the oldest languages in the top. So what do the new languages do best to be in top with the old ones, and what have the oldest to stay in top after all this time?

3. So, what programming language today?

To answer this question, we must first think about "why a language is used, chosen, learned and popular?". Being from the computer world, we could only be well positioned to react. So we determined that a popular and powerful language was a language that had learning facilities – be it semantics or structure – but also many different tools and libraries made available. We can also say that a popular language is a language that can adapt over time, whether old or new. But with new technologies coming ever faster than others, old languages – even the most powerful – have points that can be improved. And when we look at the particularity of new languages, we realize that's what they do. They are not trying to reinvent what already exists, but rather to combine everything that works best. Like Kotlin for example who is about to overtake Java for Android development. Why? Because it is easier to learn than Java, it is based on simpler semantics, which compiles for Java and JavaScript. Moreover, Kotlin does the same things as Java, or better. It is also compatible with existing Java code : it can only make you want!

Kotlin is only one example out of dozens. Other languages like Crystal that take the basics of Ruby, or Elm with JavaScript. No matter what IT domain you are programming, you will have new ways to improve your existing codes with new languages. And even if you want to stay on the languages you master, it does no matter. Just be aware that new languages – with a few exceptions like Swift – are only improvements to what you already know. If you know Java, you will have no trouble mastering Kotlin. Swift is the perfect example of a new language for a new platform. Apple wants to do everything internally, and with their huge user base, the language can only interest developers who want to reach a large number of people. If you are an application developer, you must know how to code in Java or Kotlin, but also and especially in Swift.

4. Conclusion

To conclude, nowadays you have a wide range of programming languages according to your activity, your skills and your objectives. If you already know the old languages and want to improve their productivity, do not hesitate to learn the new ones because they follow the basics you already know. If you are a beginner or a student, learning the old ones that still work today are sure values. In any case, programming is a world in perpetual evolution and the constant learning of languages should be your daily life. This can only bring you new perspectives and possibilities. Moreover, the old ones who are still present today are all as modern as the newly created languages, since a good programming language is also a language that evolve with time and new technologies.

5. References

[1] codelani.com: *"Build the next great programming language"*

[2] Wikipedia.com: *"The free encyclopedia"*

[3] tiobe.com: *"TIOBE checks more than 1056 million lines of software code for its customers "*

[4] rankred.com: *"Science, technology and knowledge"*

[5] redmonk.com: *"The developer-focused industry analyst firm"*

[6] Stack Overflow: *"Where developers learn, share & build careers"*

[7] GitHub: *"The world's leading software development platform"*