

by Java and C #, and today is a potential name: Kotlin.



1. Chinese programmers use GitHub as a 'battlefield' to oppose harsh working conditions

Programming language flow

1. Is it time to say goodbye to Java?
2. Scala or Clojure?
3. Be cautious with the choices
4. A common programming language for everyone?

Is it time to say goodbye to Java?

Back in the past a little bit, in the first issues of Technology Radar (circa 2010), the analysts touted the decline of the Java empire. Specifically, there have been alarming stagnations in the innovation problem of this long-standing programming language, making it sometimes impossible to keep up with requirements in many complex situations, especially in terms of security. Application confidentiality, and developers have begun to care about this.

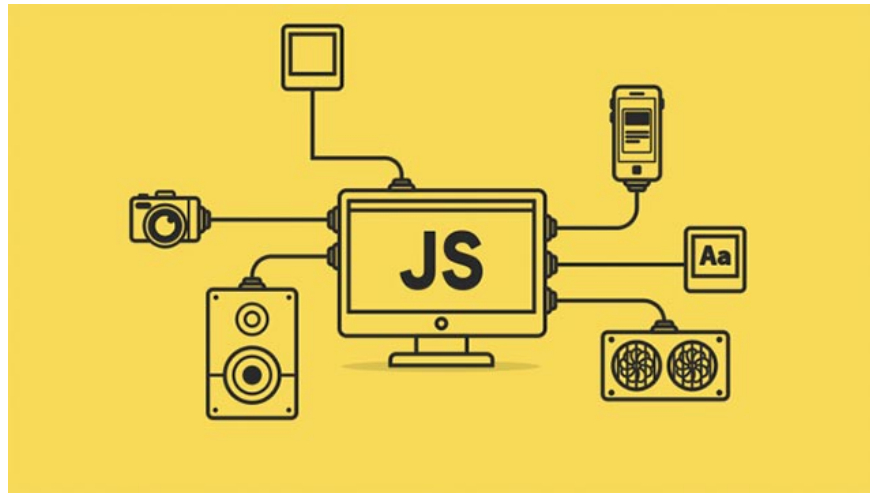


Now, after nearly 9 years, we can assume that analysts seem to have taken the problem seriously. Java has not 'died', the language still exists even though 'health' is no longer as prolific as before. But at that time, the interest of experts was very real. That's when Oracle acquired Sun microsystems and developed a new language,

inheriting the advantages of Java while also being much more powerful and superior - something we haven't witnessed in the past. raw Java.

There is one thing that experts certainly have right but it takes a lot of time for the community to catch up that is to consider JavaScript as a 'first class' language.

'First-class' languages must generally have tools that can be deployed extensively in a production environment, such as testing approaches or refactoring . In fact JavaScript has been used. for serious business-class projects really seriously. It took a bit of time, but you can see today, most people consider JavaScript an important language.



Besides, there can also be a greater interest of the community in the programming language in general. We have begun to feel the language innovations that are being built on the JVM. At that time, as a developer, you may have faced the difficult language choice to build a product, not simply 'I work in A Java environment so I will only write applications in Java ', but also about whether it is always appropriate to use another language, like Clojure or JavaScript?

1. How to 'survive' with the stress of learning programming

Scala or Clojure?


During the 20 issue of Technology Radar, analysts saw an increase in the 'functional' wave. Of course, languages like Lisp and Haskell have existed literally for decades - but what really happens is that businesses have started to become more interested in functional languages.

We can easily afford to have a situation in a programming company, with people using Clojure, while others are 'earning rice' with Scala. However, it's hard to say that between these two languages, who really wins. Both carry the power of functional languages into business - just a different approach.

Java vs Scala WordCount

```
public class WordCountJava {
    public static void main(String[] args) {
        StringTokenizer st
            = new StringTokenizer(args[0]);
        Map<String, Integer> map =
            new HashMap<String, Integer>();
        while (st.hasMoreTokens()) {
            String word = st.nextToken();
            Integer count = map.get(word);
            if (count == null)
                map.put(word, 1);
            else
                map.put(word, count + 1);
        }
        System.out.println(map);
    }
}

> runMain WordCountJava "a b a c a b"
[info] Running WordCountJava a b a c a b
{a=3, b=2, c=1}
```



```
object WordCountScala extends App {
    println(
        args(0)
            .split(" ")
            .groupBy(x => x)
            .map(t => t._1 -> t._2.length))
}

> runMain WordCountScala "a b a c a b"
[info] Running WordCountScala a b a c a b
Map(b -> 2, a -> 3, c -> 1)
```

Clojure seems to be a more pure language in a functional paradigm, while Scala tried to simplify herself using a syntax familiar to Java programmers. However, Scala is superior when it comes to its surrounding ecosystem, such as Play framework.

However, Scala's recent changes in some aspects have made it a little more complicated. New design ideas have made this language look more like Java.

This raises the question of what makes programming languages ??a good choice? It is hard to find a completely accurate answer. When choosing any language to work with, either individually or in groups, you need to determine how you will use the language, as well as how you will approach similar issues. how. For example, in a programming group, you certainly don't want a member to solve the problem in this way, while the other person has a completely different approach and implementation.

1. The 'battlefield code' and job opportunities for you

Be cautious with the choices

A clear proof for the importance of choosing a programming language is the case with Golang. This language caused quite a lot of conflicts in 2014, excitement, conflict is not missing.

GOLANG



Some famous programmers at the time did not regret the criticism and claimed that Golang was a lousy language. They think that these language creators have repeated the mistakes that have existed for decades. Meanwhile, many others say that Golang is a great invention. They love the things that this language brings to them, making it simple to believe.

This situation shows that one simple thing is that every programmer can have completely different experiences using the same language, simply because we all think about problems in other ways. together. So to know which language is right for you and your job, don't be afraid to experience it.

1. The 10 most wonderful 'shadows' of all time in the programming village

A common programming language for everyone?

As mentioned at the beginning, some people do not realize any significant value in developing new languages. This is closely linked to the idea that you should choose the only language to be a 'rice-making' tool throughout your programming career.

However, experts at ThinkWorks think this is a wrong decision. Sometimes there are things you can express more clearly in a language, while you will be struggling to do the same in another language. And one of the most important things you can do as a developer is to write the code as easily as possible.

But for programmers to freely choose and use programming languages ??on products sometimes also contains certain risks. After all, we should be cautious in expanding the options for the programming language that we or our organization will use. In order to solve different problem spaces.



1. 10 interesting facts about Python programming language

This can hardly be interpreted thoroughly, but promoting a number of languages ??that support different ecosystems or language features is very important to the business, as it allows them to speed up the process and grow faster. Meanwhile, developers also benefit by owning the right tools to solve problems at hand.

You finished reading the article "**The development flow of programming languages, new problems that are not new**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.