

# 5 interesting features about Java 9 you may not know yet

Let's TipsMake.com learn 5 interesting features about Java 9 you may not know in the article below!

1. Top 5 popular CSS Framework that you should keep in mind
2. 5 interesting features on Google Drive that users do not know
3. Instructions on how to compile and execute Java using Command Prompt

In September 2017, Java announced its latest version. After more than 3 years since Java 8, Java finally announced the next version.

**Java 9** is a turning point in Java's release life cycle. In this article, we will discuss new developer features in Java 9. From an operational perspective, there are changes and improvements in performance and security. After developing Java 9 for months at ValueCoders, we'll talk about the five most interesting features in Java 9 to encourage you to apply it.

Let's TipsMake.com learn **5 interesting features about Java 9 you may not know** in the article below!

## 1. Modules

Let's start with the most important feature of Java 9 - Java Platform Module System. The biggest problem with Java is the library runtime. Twenty years of development on a large platform has created a lot of confusion with classes overlapping. Even, there is no optimal way to deploy classes in the JDK. That led to the use of external libraries, depending on the details inside, making the development of the Java platform hampered.

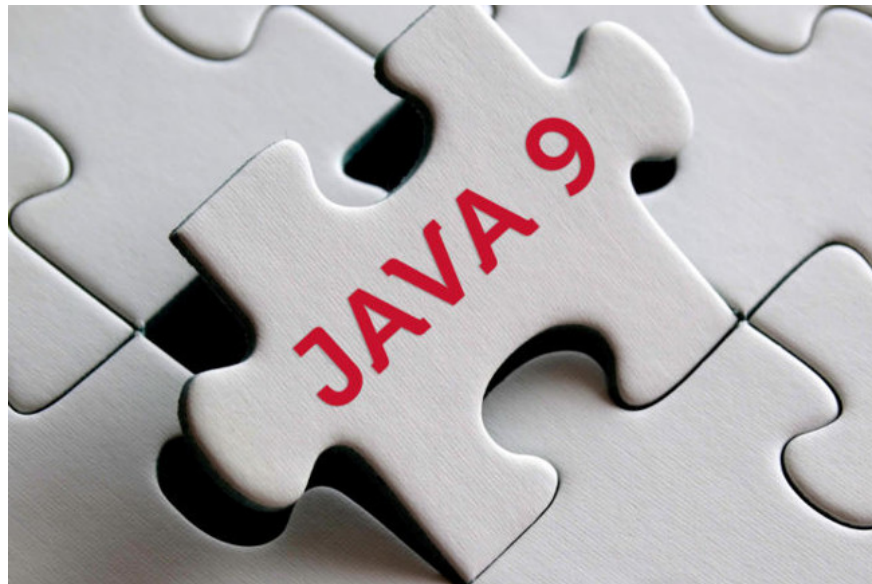


With Java 9, the JDK itself is modular and comes with 90 other separate modules. Only packages that are explicitly exported from the module can be accessed to other modules. Finally, internal implementation classes can now encapsulate users' information with extra information fields, to perform network communication. In addition, modules can express a clear dependence on other modules. Dependencies are always checked by the module system both during compilation and execution.

Modules can also be used to modular applications, not just JDKs. Instead of using fragile classpath, you can now create your own modules with explicit packages and attributes.

The module provides a path to applications with a clearer and more reliable structure. However, this new module system will not change Java in a short time. Modules affect your application design and manifest themselves in the development stages: development, compilation, packaging and implementation. It will take some time before the modular system is widely applied. Fortunately, the community welcomes this change very positively.

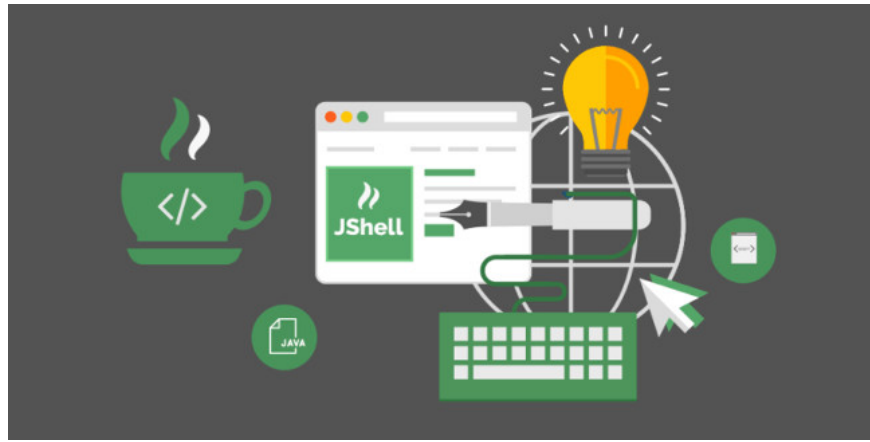
## 2. Productivity



In addition to the modular system, Java 9 also has small features to improve developer productivity. For example, collection factory methods allow you to create immutable collections with predefined elements in a statement. Thus there is no need to deal with more troubles to create small, predefined collections.

Another key productivity acceleration feature is **JShell**, a Read-Eval-Print-Loop code for Java. Instead of compiling and running small classes to test code, you can interactively import and evaluate Java code snippets in JShell with instant feedback. JShell has been integrated with IDEs such as IntelliJ IDEA, allowing you to open an interactive environment with access to your application code.

## 3. Performance



Running a new version of Java means you will get performance improvements without changing the code. Java 9 is no exception. An example of such an improvement is the Compact Strings feature. Any string can be seen only with ASCII character set, storing one byte for each character. Strings with characters outside the ASCII range will take two bytes for each character. Most applications have a good amount of String objects at work. If they are simple strings, this will reduce the amount of memory used.

Another important change in Java 9 is moving to Garbage Collector G1 (Garbage First) as the garbage collector set by default. Designed for large-scale use, it provides adjustable time off targets and increased collection performance. The impact of this change depends on your application characteristics.

In general, you will get lower latency when executing code. In return, G1GC uses more CPU resources to achieve this. However, it is important to check garbage collector behavior on specific applications when switching to Java 9. In some cases, the allocation patterns do not match the G1GC approach and the switch to GC Other (or G1GC) is needed.

## 4. Security



The productivity features of developers are not the only reason to apply Java 9. The platform itself has been updated with many new features to enhance security. Most security enhancements in Java 9 relate to security networks. The latest versions of (D) TLS are currently supported for secure HTTPS connections.

Other security enhancements include the ability to filter sequential data. Sequencing is something that Java has always overlooked, leading to many security attacks in recent years. Now, you can filter sequential data streams to help reduce the level of Java serialization attacks.

## 5. Release cycle

```
import java.util.Random;
import java.util.stream.Collectors;
import java.util.stream.Stream;

public class Employee {
    private String name;
    private int salary;
}
```

In the previous section, we only focus on the features of Java 9. But there is another reason to prove that Java 9 is an important version: It marks the end of the era of Java-only versions centered on 'features'. In the past, some important features were chosen and when ready, a new version of Java was created. In fact, this leads to a gap of at least three years between the major versions. This is inherently unacceptable for today's technological age. Therefore, Oracle and OpenJDK are moving to releasing timeline-based versions.

Now, every 6 months, a new version of Java will be released. It will contain all the features available at that time. Between these 6-month releases, the company will also release an update with important fixes if needed. This new release schedule has some important implications. Java versions become popular and predictable. Currently Java will release major updates in March and September every year. As planned, the new version will be released every 6 months, which means that the next version in March 2018 will be Java 10. However, the company has not released official information to confirm the timeline. point on.

However, updating Java regularly is not good news for many companies. Therefore, there will be long-term versions (LTSs) designated every three years. The first version of LTS will be released in September 2018. This also means that Java 9 will not be included in supported long-term versions.

### Epilogue

These new strategies make Java more flexible than ever. The main features will be divided into small steps and updated every 6 months. So even after more than 20 years, Java continues to grow at a fast and rapid pace.

Refer to some more articles:

1. Basic Java syntax
2. Download and install Java on the computer
3. 12 extremely useful tricks for JavaScript programmers

Having fun!

You finished reading the article "**5 interesting features about Java 9 you may not know yet**" 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.

---