**推荐！国外程序员整理的Java资源大全**

**构建**

这里搜集了用来构建应用程序的工具。

* [Apache Maven](http://maven.apache.org/)：Maven使用声明进行构建并进行依赖管理，偏向于使用约定而不是配置进行构建。Maven优于Apache Ant。后者采用了一种过程化的方式进行配置，所以维护起来相当困难。
* [Gradle](http://www.gradle.org/)：Gradle采用增量构建。Gradle通过Groovy编程而不是传统的XML声明进行配置。Gradle可以很好地配合Maven进行依赖管理，并且把Ant脚本当作头等公民。

**字节码操作**

编程操作Java字节码的函数库。

* [ASM](http://asm.ow2.org/)：通用底层字节码操作及分析。
* [Javassist](http://www.csg.ci.i.u-tokyo.ac.jp/%7Echiba/javassist/)：尝试简化字节码编辑。
* [Byte Buddy](http://bytebuddy.net/)：使用“流式API”进一步简化字节码生成。

**代码分析**

软件度量和质量评估工具。

* [Checkstyle](http://checkstyle.sourceforge.net/)：对编程规范和标准进行静态分析。
* [FindBugs](http://findbugs.sourceforge.net/)：通过字节码静态分析找出潜在Bug。
* [PMD](http://pmd.sourceforge.net/)：对源代码中不良编程习惯进行分析。
* [SonarQube](http://www.sonarqube.org/)：通过插件集成其它分析组件，提供评估最终结果报告。

**编译器**

创建分析器、解释器和编译器的框架。

* [ANTLR](http://www.antlr.org/)：功能完备的自顶向下分析复杂框架。
* [JavaCC](https://javacc.java.net/)：相对ANTLR更具体，上手略为简单。支持语法语法超前预测（syntactic lookahead）。

**持续集成**

支持持续集成、测试和应用发布的工具。

* [Bamboo](https://www.atlassian.com/software/bamboo)：Atlassian的持续集成（CI）解决方案，包含很多其它产品。
* [CircleCI](https://circleci.com/)：提供托管服务，可免费试用。
* [Codeship](https://www.codeship.io/features)：提供托管服务，提供有限免费计划。
* [Go](http://www.thoughtworks.com/products/go-continuous-delivery)：ThoughtWork开源持续集成解决方案。
* [Jenkins](http://jenkins-ci.org/)：提供基于服务器的部署服务。
* [TeamCity](http://www.jetbrains.com/teamcity/)：JetBrain持续集成方案，提供免费版。
* [Travis](https://travis-ci.org)：提供托管服务，常用于开源项目。

**数据库**

简化数据库交互的工具、库。

* [Flyway](http://flywaydb.org/)：使用Java API轻松完成数据库迁移。
* [H2](http://h2database.com/)：小型SQL数据库，以内存操作著称。
* [JDBI](http://jdbi.org/)：便捷的JDBC抽象。
* [jOOQ](http://www.jooq.org/)：基于SQL schema生成类型安全代码。
* [Presto](https://github.com/facebook/presto)：针对大数据的分布式SQL查询引擎。
* [Querydsl](http://www.querydsl.com/)：针对Java的类型安全统一查询。

**日期和时间**

处理日期和时间的函数库。

* [Joda-Time](http://www.joda.org/joda-time/)：Java 8出现之前，它是日期、时间处理的标准函数库。
* [Time4J](https://github.com/MenoData/Time4J)：Java高级日期、时间函数库。

**依赖注入**

帮助代码实现[控制反转](http://en.wikipedia.org/wiki/Inversion_of_control)模式的函数库。

* [Dagger](http://square.github.io/dagger/) ：编译期的注入框架，没有使用反射，主要用于Android开发。
* [Guice](https://github.com/google/guice)：轻量级注入框架，功能强大可与Dagger媲美。

**开发库**

从基础层次上改进开发流程。

* [AspectJ](https://eclipse.org/aspectj/)：面向切面编程扩展，与程序无缝连接。
* [Auto](https://github.com/google/auto)：源代码生成器集合。
* [DCEVM](http://ssw.jku.at/dcevm/)：通过修改JVM，在运行时可无限次重定义已加载的类。OpenJDK 7、8已提供支持，详情可查看[这个分支（fork）](http://dcevm.github.io/)。
* [JRebel](http://zeroturnaround.com/software/jrebel/)：商用软件，无需重新部署可即时重新加载代码及配置。
* [Lombok](http://projectlombok.org/)：代码生成器，旨在减少Java冗余代码。
* [RxJava](https://github.com/Netflix/RxJava)：使用JVM中可观察序列，创建异步、基于事件应用程序的函数库。
* [Spring Loaded](https://github.com/spring-projects/spring-loaded)：另一个JVM类重载代理。
* [vert.x](http://vertx.io/)：JVM多语言事件驱动应用框架。

**分布式应用**

用来开发分布式、具有容错性应用程序的函数库和框架。

* [Akka](http://akka.io)：构建并发、分布式和具有容错功能的事件驱动应用程序所需的工具包和运行时。
* [Apache Storm](http://storm.incubator.apache.org/)：分布式实时计算系统。
* [Apache ZooKeeper](http://zookeeper.apache.org/)：为大型分布式系统，使用分布式配置、同步和命名注册提供协调服务。
* [Hazelcast](http://hazelcast.org/)：分布式、高可扩展性内存网格。
* [Hystrix](https://github.com/Netflix/Hystrix)：为分布式系统提供延迟和容错处理。
* [JGroups](http://www.jgroups.org/)：一组提供可靠消息传输的工具包，可用来创建集群。集群中的节点可互相发送消息。
* [Quasar](http://www.paralleluniverse.co/quasar/)：为JVM提供轻量级线程和Actor。

**发布**

使用本机格式分发Java应用程序的工具。

* [Bintray](https://bintray.com/)：对二进制发布进行版本控制，可与Maven或Gradle配合使用。
* [IzPack](http://izpack.org/)：为跨平台部署建立授权工具。
* [Launch4j](http://launch4j.sourceforge.net/)：将JAR包装为小巧的Windows可执行文件。
* [packr](https://github.com/libgdx/packr#packr)：将程序JAR、资源和JVM打包成Windows、Linux和Mac OS X的本机文件。

**文档处理**

用来处理Office格式文档的函数库。

* [Apache POI](http://poi.apache.org/)：支持OOXML （XLSX、DOCX、PPTX）以及 OLE2 （XLS, DOC or PPT）格式的文档。
* [jOpenDocument](http://www.jopendocument.org/)：处理OpenDocument格式文档。

**游戏开发**

游戏开发框架。

* [jMonkeyEngine](http://jmonkeyengine.org/)：支持现代3D开发的游戏引擎。
* [libGDX](http://libgdx.badlogicgames.com/)：全面的跨平台高级开发框架。
* [LWJGL](http://lwjgl.org/)：抽象了OpenGL、CL、AL等函数库的健壮框架。

**GUI**

用来创建现代图形用户界面的函数库。

* [JavaFX](http://www.oracle.com/technetwork/java/javase/overview/javafx-overview-2158620.html)：Swing的继承者。
* [Scene Builder](http://www.oracle.com/technetwork/java/javase/downloads/javafxscenebuilder-info-2157684.html)：JavaFX虚拟布局工具。

**高性能**

与高性能计算有关的资源，包括集合以及很多具体功能的函数库。

* [Disruptor](http://lmax-exchange.github.io/disruptor/)：线程间消息函数库。
* [fastutil](http://fastutil.di.unimi.it/)：快速紧凑的Java类型安全集合。
* [GS Collections](https://github.com/goldmansachs/gs-collections)：受Smalltalk启发的集合框架。
* [hftc](https://github.com/OpenHFT/hftc)：Hash set和hash map。
* [HPPC](http://labs.carrotsearch.com/hppc.html)：基本类型集合。
* [Javolution](http://javolution.org/)：针对实时嵌入式系统的函数库。
* [Trove](http://trove.starlight-systems.com/)：基本类型集合。

**IDE**

视图简化开发的集成开发环境。

* [Eclipse](http://www.eclipse.org/)：后台做了很多工作，以其丰富插件著称。
* [IntelliJ IDEA](http://www.jetbrains.com/idea/)：支持很多JVM语言，为Android开发提供了很多不错的选项。其商业版本主要面向企业用户。
* [NetBeans](https://netbeans.org/)：集成了很多Java SE和Java EE特性，包括数据库访问、服务器、HTML5以及AngularJS。

**图像处理**

用来帮助创建、评估或操作图形的函数库。

* [Picasso](http://square.github.io/picasso/)：Android下载图像和图像缓存函数库。
* [ZXing](https://github.com/zxing/zxing)：多种格式的一维、二维条形码处理函数库。

**JSON**

简化JSON处理的函数库。

* [Gson](https://code.google.com/p/google-gson/)：将Java对象序列化为JSON及反向操作。使用时提供了很好的性能。
* [Jackson](http://wiki.fasterxml.com/JacksonHome)：与GSON类似，但如果需要频繁初始化Jackson库会带来性能问题。

**JVM和JDK**

目前的JVM、JDK实现。

* [JDK 9](https://jdk9.java.net/)：JDK 9早期访问版本。
* [OpenJDK](http://openjdk.java.net/)：开源实现。

**日志**

记录应用程序的日志函数库。

* [Apache Log4j 2](http://logging.apache.org/log4j/)：对之前版本进行了完全重写。现在的版本具备一个强大的插件和配置架构。
* [kibana](http://www.elasticsearch.org/overview/kibana/)：对日志进行分析并进行可视化。
* [Logback](http://logback.qos.ch/)：Log4j原班人马作品。被证明是一个强健的日志函数库，通过Groovy提供了很多有意思的配置选项。
* [logstash](http://logstash.net/)：日志文件管理工具。
* [SLF4J](http://www.slf4j.org/)：日志抽象层，需要与某个具体日志框架配合使用。

**机器学习**

提供具体统计算法的工具。其算法可从数据中学习。

* [Apache Hadoop](http://hadoop.apache.org/)：对商用硬件集群上大规模数据存储和处理的开源软件框架。
* [Apache Mahout](https://mahout.apache.org/)：专注协同过滤、聚类和分类的可扩展算法。
* [Apache Spark](http://spark.apache.org/)：开源数据分析集群计算框架。
* [h2o](http://0xdata.github.io/h2o/)：用作大数据统计的分析引擎。
* [Weka](http://www.cs.waikato.ac.nz/ml/weka/)：用作数据挖掘的算法集合，包括从预处理到可视化的各个层次。

**消息**

在客户端之间进行消息传递，确保协议独立性的工具。

* [Apache ActiveMQ](http://activemq.apache.org/)：实现JMS的开源消息代理（broker），可将同步通讯转为异步通讯。
* [Apache Kafka](http://kafka.apache.org/)：高吞吐量分布式消息系统。
* [JBoss HornetQ](http://hornetq.jboss.org/)：清晰、准确、模块化且方便嵌入的消息工具。
* [JeroMQ](https://github.com/zeromq/jeromq)：ZeroMQ的纯Java实现。

**其它**

其它资源。

* [Design Patterns](https://github.com/iluwatar/java-design-patterns)：实现并解释了最常见的[设计模式](http://www.amazon.cn/gp/product/B001130JN8/ref=as_li_qf_sp_asin_il_tl?ie=UTF8&tag=importnew-23&linkCode=as2&camp=536&creative=3200&creativeASIN=B001130JN8)。
* [Jimfs](https://github.com/google/jimfs)：内存文件系统。
* [Lanterna](https://code.google.com/p/lanterna/)：类似curses的简单console文本GUI函数库。
* [LightAdmin](http://lightadmin.org/)：可插入式CRUD UI函数库，可用于快速应用开发。
* [Metrics](http://metrics.codahale.com/)：创建自己的软件度量或者为支持框架添加度量信息，通过JMX或HTTP进行发布或者发送到数据库。
* [OpenRefine](http://openrefine.org/)：用来处理混乱数据的工具，包括清理、转换、使用Web Service进行扩展并将其关联到数据库。
* [RoboVM](http://www.robovm.org/)：Java编写原生iOS应用。

**自然语言处理**

用来专门处理文本的函数库。

* [Apache OpenNL](https://opennlp.apache.org/)：处理类似分词等常见任务的工具。
* [CoreNLP](http://nlp.stanford.edu/software/corenlp.shtml)：斯坦佛的CoreNLP提供了一组基础工具，可以处理类似标签、实体名识别和情感分析这样的任务。
* [LingPipe](http://alias-i.com/lingpipe/)：一组可以处理各种任务的工具集，支持POS标签、情感分析等。
* [Mallet](http://mallet.cs.umass.edu/)：统计学自然语言处理、文档分类、聚类、主题建模等。

**网络**

网络编程函数库。

* [Netty](http://netty.io/)：构建高性能网络应用程序开发框架。
* [OkHttp](http://square.github.io/okhttp/) ：一个Android和Java应用的HTTP+SPDY客户端。

**ORM**

处理对象持久化的API。

* [EclipseLink](https://www.eclipse.org/eclipselink/)：支持许多持久化标准，JPA、JAXB、JCA和SDO。
* [Hibernate](http://hibernate.org/orm/)：广泛使用、强健的持久化框架。Hibernate的技术社区非常活跃。
* [Ebean](http://ebean-orm.github.io/)：支持快速数据访问和编码的ORM框架。

**PDF**

用来帮助创建PDF文件的资源。

* [Apache FOP](http://xmlgraphics.apache.org/fop/)：从XSL-FO创建PDF。
* [Apache PDFBox](http://pdfbox.apache.org/)：用来创建和操作PDF的工具集。
* [DynamicReports](http://dynamicreports.org/)：JasperReports的精简版。
* [iText](http://itextpdf.com/)：一个易于使用的PDF函数库，用来编程创建PDF文件。注意，用于商业用途时需要许可证。
* [JasperReports](http://community.jaspersoft.com/project/jasperreports-library)：一个复杂的报表引擎。

**REST框架**

用来创建RESTful 服务的框架。

* [Dropwizard](https://dropwizard.github.io/dropwizard/)：偏向于自己使用的Web框架。用来构建Web应用程序，使用了Jetty、Jackson、Jersey和Metrics。
* [Jersey](https://jersey.java.net/)：JAX-RS参考实现。
* [RESTEasy](http://resteasy.jboss.org/)：经过JAX-RS规范完全认证的可移植实现。
* [Retrofit](http://square.github.io/retrofit/)：一个Java类型安全的REST客户端。
* [Spark](http://www.sparkjava.com/)：受到Sinatra启发的Java REST框架。
* [Swagger](https://helloreverb.com/developers/swagger)：Swagger是一个规范且完整的框架，提供描述、生产、消费和可视化RESTful Web Service。

**科学**

用于科学计算和分析的函数库。

* [SCaVis](http://jwork.org/scavis/)：用于科学计算、数据分析和数据可视化环境。

**搜索**

文档索引引擎，用于搜索和分析。

* [Apache Solr](http://lucene.apache.org/solr/) ：一个完全的企业搜索引擎。为高吞吐量通信进行了优化。
* [Elasticsearch](http://www.elasticsearch.org/)：一个分布式、支持多租户（multitenant）全文本搜索引擎。提供了RESTful Web接口和无schema的JSON文档。

**安全**

用于处理安全、认证、授权或会话管理的函数库。

* [Apache Shiro](http://shiro.apache.org/)：执行认证、授权、加密和会话管理。
* [Cryptomator](http://cryptomator.org/)：在云上进行客户端跨平台透明加密。
* [Keycloak](http://keycloak.jboss.org/)：为浏览器应用和RESTful Web Service集成SSO和IDM。目前还处于beta版本，但是看起来非常有前途。
* [PicketLink](http://picketlink.org/)：PicketLink是一个针对Java应用进行安全和身份认证管理的大型项目（Umbrella Project）。
* [Spring Security](http://projects.spring.io/spring-security/)：专注认证、授权和多维度攻击防护框架。

**序列化**

用来高效处理序列化的函数库。

* [FlatBuffers](https://github.com/google/flatbuffers)：序列化函数库，高效利用内存，无需解包和解析即可高效访问序列化数据。
* [Kryo](https://github.com/EsotericSoftware/kryo)：快速和高效的对象图形序列化框架。
* [MessagePack](https://github.com/msgpack/msgpack-java)：一种高效的二进制序列化格式。

**服务器**

用来部署应用程序的服务器。

* [Apache Tomcat](http://tomcat.apache.org/)：针对Servlet和JSP的应用服务器，健壮性好且适用性强。
* [Apache TomEE](http://tomee.apache.org/)：Tomcat加Java EE。
* [GlassFish](https://glassfish.java.net/)：Java EE开源参考实现，由Oracle资助开发。
* [Jetty](http://www.eclipse.org/jetty/)：轻量级、小巧的应用服务器，通常会嵌入到项目中。
* [WildFly](http://www.wildfly.org/)：之前被称作JBoss，由Red Hat开发。支持很多Java EE功能。

**模版引擎**

对模板中表达式进行替换的工具。

* [Apache Velocity](http://velocity.apache.org/)：提供HTML页面模板、email模板和通用开源代码生成器模板。
* [FreeMarker](http://freemarker.org/)：通用模板引擎，不需要任何重量级或自己使用的依赖关系。
* [Handlebars.java](http://jknack.github.io/handlebars.java/)：使用Java编写的模板引擎，逻辑简单，支持语义扩展（semantic Mustache）。
* [JavaServer Pages](https://jsp.java.net/)：通用网站模板，支持自定义标签库。
* [Thymeleaf](http://www.thymeleaf.org/)：旨在替换JSP，支持XML文件。

**测试**

测试内容从对象到接口，涵盖性能测试和基准测试工具。

* [Apache JMeter](http://jmeter.apache.org/)：功能性测试和性能评测。
* [Arquillian](http://arquillian.org/)：集成测试和功能行测试平台，集成Java EE容器。
* [AssertJ](http://joel-costigliola.github.io/assertj/)：支持流式断言提高测试的可读性。
* [JMH](http://openjdk.java.net/projects/code-tools/jmh/)：JVM微基准测试工具。
* [JUnit](http://junit.org/)：通用测试框架。
* [Mockito](http://code.google.com/p/mockito/)：在自动化单元测试中创建测试对象，为TDD或BDD提供支持。
* [Selenium](http://docs.seleniumhq.org/)：为Web应用程序提供可移植软件测试框架。
* [Selenide](http://selenide.org/)：为Selenium提供精准的周边API，用来编写稳定且可读的UI测试。
* [TestNG](http://testng.org/) ：测试框架。
* [VisualVM](http://visualvm.java.net/)：提供可视化方式查看运行中的应用程序信息。

**工具类**

通用工具类函数库。

* [Apache Commons](http://commons.apache.org/)：提供各种用途的函数，比如配置、验证、集合、文件上传或XML处理等。
* [Guava](http://code.google.com/p/guava-libraries/)：集合、缓存、支持基本类型、并发函数库、通用注解、字符串处理、I/O等。
* [javatuples](http://www.javatuples.org/)：正如名字表示的那样，提供tuple支持。尽管目前tuple的概念还有留有争议。

**网络爬虫**

用于分析网站内容的函数库。

* [Apache Nutch](http://nutch.apache.org/) ：可用于生产环境的高度可扩展、可伸缩的网络爬虫。
* [Crawler4j](https://code.google.com/p/crawler4j/)：简单的轻量级爬虫。
* [JSoup](http://jsoup.org/) ：刮取、解析、操作和清理HTML。

**Web框架**

用于处理Web应用程序不同层次间通讯的框架。

* [Apache Tapestry](http://tapestry.apache.org/)：基于组件的框架，使用Java创建动态、强健的、高度可扩展的Web应用程序。
* [Apache Wicket](http://wicket.apache.org/)：基于组件的Web应用框架，与Tapestry类似带有状态显示GUI。
* [Google Web Toolkit](http://www.gwtproject.org/)：一组Web开发工具集，包含在客户端将Java代码转为JavaScript的编译器、XML解析器、RCP API、JUnit集成、国际化支持和GUI控件。
* [Grails](https://grails.org/)：Groovy框架，旨在提供一个高效开发环境，使用约定而非配置、没有XML并支持混入（mixin）。
* [Play](http://www.playframework.com/)： 使用约定而非配置，支持代码热加载并在浏览器中显示错误。
* [PrimeFaces](http://primefaces.org/)：JSF框架，提供免费版和带技术支持的商业版。包含一些前端组件。
* [Spring Boot](http://projects.spring.io/spring-boot/)：微框架，简化了Spring新程序的开发过程。
* [Spring](http://projects.spring.io/spring-framework/)：旨在简化Java EE的开发过程，提供依赖注入相关组件并支持面向切面编程。
* [Vaadin](https://vaadin.com/)：基于GWT构建的事件驱动框架。使用服务端架构，客户端使用Ajax。
* [Ninja](http://www.ninjaframework.org/)：Java全栈Web开发框架。非常稳固、快速和高效。
* [Ratpack](http://www.ratpack.io/)：一组Java开发函数库，用于构建快速、高效、可扩展且测试完备的HTTP应用程序。

**资源**

**社区**

活跃的讨论区。

* [r/java](http://www.reddit.com/r/java)：Java社区的Subreddit。
* [stackoverflow](http://stackoverflow.com/questions/tagged/java)：问答平台。

**有影响的书籍**

具有广泛影响且值得阅读的Java经典书籍。

* [Effective Java (2nd Edition)](http://www.amazon.cn/gp/product/0321356683/ref=as_li_qf_sp_asin_il_tl?ie=UTF8&camp=536&creative=3200&creativeASIN=0321356683&linkCode=as2&tag=vastwork-23)
* [Java Concurrency in Practice](http://www.amazon.com/Java-Concurrency-Practice-Brian-Goetz/dp/0321349601) | [Java并发编程实战](http://www.amazon.cn/gp/product/B0077K9XHW/ref=as_li_qf_sp_asin_il_tl?ie=UTF8&camp=536&creative=3200&creativeASIN=B0077K9XHW&linkCode=as2&tag=vastwork-23)
* [Thinking in Java](http://www.amazon.cn/gp/product/B0011C3CVA/ref=as_li_qf_sp_asin_il_tl?ie=UTF8&camp=536&creative=3200&creativeASIN=B0011C3CVA&linkCode=as2&tag=vastwork-23) | [中文版](http://www.amazon.cn/gp/product/B0011F7WU4/ref=as_li_qf_sp_asin_il_tl?ie=UTF8&camp=536&creative=3200&creativeASIN=B0011F7WU4&linkCode=as2&tag=vastwork-23)

**播客**

可以一边编程一边听的东西。

* [The Java Posse](http://www.javaposse.com/)

**Twitter**

值得关注的帐号。

* [Adam Bien](https://twitter.com/AdamBien/)：自由职业者、作家、JavaONE明星演讲者、顾问、Java Champion。
* [Antonio Goncalves](https://twitter.com/agoncal/)：Java Champion、JUG Leader、Devoxx France、Java EE 6/7、JCP、作家。
* [Arun Gupta](https://twitter.com/arungupta/)：Java Champion、JavaONE明星演讲者、JUG Leader、Devoxx4Kids成员、Red Hatter。
* [Bruno Borges](https://twitter.com/brunoborges)：Oracle产品经理、Java Jock。
* [Ed Burns](https://twitter.com/edburns)：Oracle技术团队顾问。
* [Eugen Paraschiv](https://twitter.com/baeldung)：Spring安全课程作者。
* [James Weaver](https://twitter.com/JavaFXpert)：Java、JavaFX、IoT开发者、作者和演讲者。
* [Java EE](https://twitter.com/Java_EE/)：Java EE Twitter官方账号。
* [Java Magazine](https://twitter.com/Oraclejavamag)：Java杂志官方账号。
* [Java.net](https://twitter.com/javanetbuzz/)：Java.net官方账号。
* [Java](https://twitter.com/java/)：Java Twitter官方账号。
* [Javin Paul](https://twitter.com/javinpaul)：知名Java博客作者。
* [Lukas Eder](https://twitter.com/lukaseder)：Data Geekery（jOOQ）创始人兼CEO。
* [Mario Fusco](https://twitter.com/mariofusco)：RedHatter、JUG协调、活跃讲师和作者。
* [Mark Reinhold](https://twitter.com/mreinhold)：Oracle首席架构师、Java平台开发组。
* [Martijn Verburg](https://twitter.com/karianna)：London JUG co-leader、演讲者、作家、Java Champion等。
* [OpenJDK](https://twitter.com/OpenJDK)：OpenJDK官方账号。
* [Reza Rahman](https://twitter.com/reza_rahman)：Java EE、GlassFish、WebLogic传道者、作家、演讲者、开源黑客。
* [Simon Maple](https://twitter.com/sjmaple)：Java Champion、virtualJUG创始人、LJC leader、RebelLabs作者。
* [Stephen Colebourne](https://twitter.com/jodastephen)： Java Champion、演讲者。
* [Tim Boudreau](https://twitter.com/kablosna)：作家、NetBeans大牛。
* [Trisha Gee](https://twitter.com/trisha_gee)：Java Champion、演讲者。

**网站**

值得阅读的网站。

* [Google Java Style](http://google-styleguide.googlecode.com/svn/trunk/javaguide.html)
* [InfoQ](http://www.infoq.com/)
* [Java Code Geeks](http://www.javacodegeeks.com/)
* [Java.net](http://java.net/)
* [Javalobby](http://java.dzone.com/)
* [JavaWorld](http://www.javaworld.com/)
* [RebelLabs](http://zeroturnaround.com/rebellabs/)
* [The Java Specialist’ Newsletter](http://www.javaspecialists.eu/archive/archive.jsp)
* [TheServerSide.com](http://www.theserverside.com/)
* [Thoughts On Java](http://www.thoughts-on-java.org/)
* [ImportNew](http://www.importnew.com)（ImportNew 专注 Java 技术）

**参与贡献**

热烈欢迎参与贡献此列表！

请参阅[CONTRIBUTING](https://github.com/akullpp/awesome-java/blob/master/CONTRIBUTING.md)加入贡献。

**可能感兴趣的文章**

* [5种调优Java NIO和NIO.2的方式](http://www.importnew.com/2000.html)
* [Google I/O 2013大会Andoird更新前瞻](http://www.importnew.com/4157.html)
* [Eclipse Kepler有关的数据](http://www.importnew.com/5702.html)
* [你不需要 Hadoop做数据分析的10个理由 —— 使用之前必须测试其他替代品](http://www.importnew.com/5759.html)
* [CSRF 攻击的应对之道](http://www.importnew.com/5839.html)
* [类加载器的工作原理](http://www.importnew.com/6581.html)
* [equals与”==”的区别](http://www.importnew.com/6804.html)
* [Java垃圾回收精粹 — Part2](http://www.importnew.com/8343.html)
* [Java反编译器剖析（下）](http://www.importnew.com/9248.html)
* [自恢复JVM](http://www.importnew.com/14137.html)