

Gradle&Groovy学习

简单的说，Gradle是一个构建工具，它是用来帮助我们构建app的，构建包括编译、打包等过程。我们可以为Gradle指定构建规则，然后它就会根据我们的“命令”自动为我们构建app。

目录结构

build.gradle

代表了app Module的构建脚本，它定义了应用于本模块的构建规则

gradle.properties

这个文件中定义了一系列供build.gradle使用的常量，比如keystore的存储路径、keyalias等。

gradlew & gradlew.bat

gradlew为Linux下的shell脚本，gradlew.bat是Windows下的批处理文件。gradlew是gradle wrapper的缩写，也就是说它对gradle的命令进行了包装，比如我们进入到指定Module目录并执行“gradlew.bat assemble”即可完成对当前Module的构建（Windows系统下）。

local.properties

这个文件中定义了一些本地属性，比如SDK的路径。

setting.gradle

假如我们的项目包含了不只一个Module时，我们想要一次性构建所有Module以完成整个项目的构建，这时我们需要用到这个文件。

gradle wrapper

Wrapper是对Gradle的一层包装,便于在团队开发过程中统一Gradle构建的版本。

Wrapper启动Gradle时会检查Gradle有没有被下载关联，若没有就会从配置的地址下载并运行构建。

Groovy

Groovy是一种运行在JVM上的动态语言，它吸取了Python,Ruby和Smalltalk等语言的优点，

在Java语言的基础之上增加了许多特色功能；对于Java开发人员来说掌握Groovy是没有什么太大障碍的；

相比Java而言，语法更易懂，更易上手，更易调试；无缝的集成了Java的类和库；编译后的.groovy也是以class的形式出现的。

Groovy 是用于Java虚拟机的一种敏捷的动态语言，它是一种成熟的面向对象编程语言，既可以用于面向对象编程，又可以用作纯粹的脚本语言。使用该种语言不必编写过多的代码，同时又具有闭包和动态语言中的其他特性。

Groovy是JVM的一个替代语言（替代是指可以用 Groovy 在Java平台上进行 Java 编程），使用方式基本与使用 Java代码的方式相同，该语言特别适合与Spring的动态语言支持一起使用，设计时充分考虑了Java集成，这使 Groovy 与 Java 代码的互操作很容易。

环境配置：

下载groovy

<http://groovy.codehaus.org/Download>

在系统变量中配置groovy环境变量，并添加到PATH中

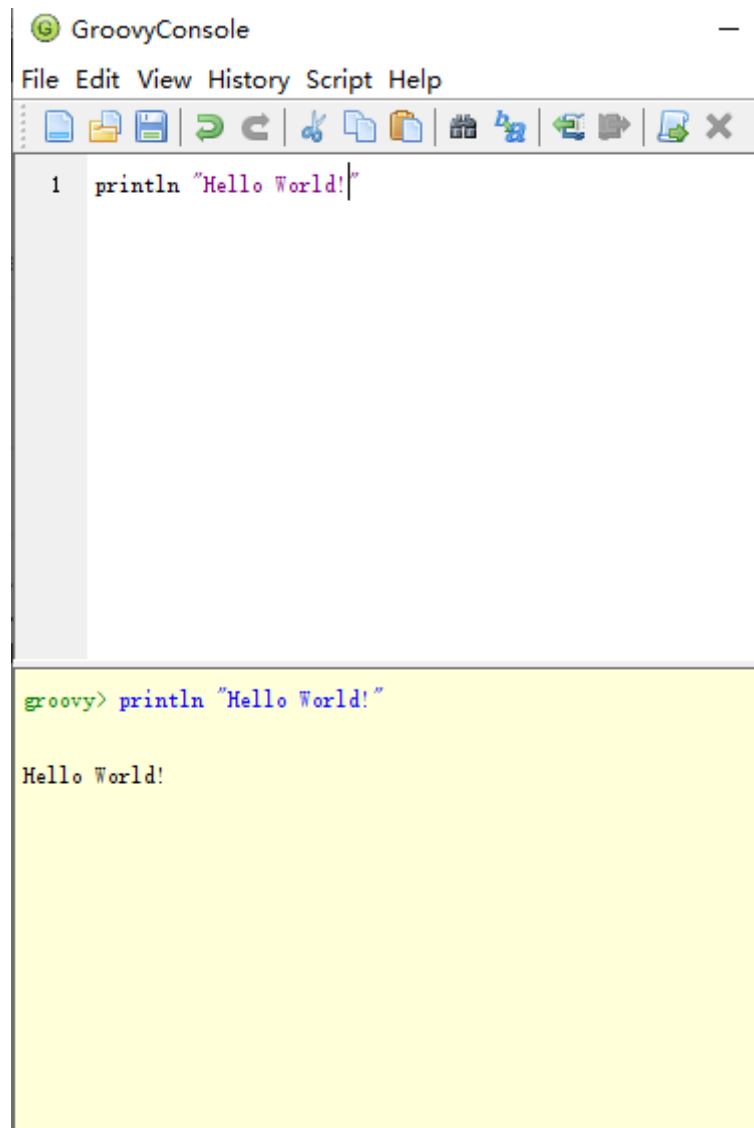
命令行键入groovy -version

```
C:\Users\dell>groovy -version
Groovy Version: 2.3.6 JVM: 1.8.0_144 Vendor: Oracle Corporation OS: Windows 10
```

命令行键入groovyconsole进入自带编辑器

```
C:\Users\dell>groovyconsole
四月 03, 2019 10:28:43 上午 java.util.prefs.WindowsPreferences <init>
警告: Could not open/create prefs root node Software\JavaSoft\Prefs at root 0x80
000002. Windows RegCreateKeyEx(...) returned error code 5.
```

输入相关指令，按CTRL + R运行



也可在IDEA中直接编辑groovy，创建一个groovy对象即可