Gradle Guide

* 开始

1.安装Gradle(以windows为例)

1.1.下载Gradle，解压，配置环境变量到PATH

1.2.使用Android Studio，可以使用Gradle Wrapper。在**gradle/wrapper/gradle-wrapper.properties**路径下，更改**distributionUrl**的值，获取不同版本的**Gradle Wrapper.**创建

**2.**新的**Gradle**构建

初始化项目

**>mkdir gradle-demo**

**>cd gradle-demo**

**>gradle init**

目录结构：.

├── build.gradle 1

├── gradle

│   └── wrapper

│   ├── gradle-wrapper.jar 2

│   └── gradle-wrapper.properties 3

├── gradlew 4

├── gradlew.bat 5

└── settings.gradle 6

**1.project**的配置脚本，用来配置任务

**2.Gradle Wrapper**的执行**jar**包

**3.Gradle Wrapper**属性配置文件

**4.Unix-based** 操作系统的**Gradle Wrapper**脚本

**5.Windows**操作系统的**Gradle Wrapper**脚本

**6.Settings**配置脚本，用来配置**Project**中参与编译的模块

3.任务

task copy(type: Copy, group: "Custom", description: "Copies sources to the dest directory") {

from "src"

into "dest"

}

>./gradlew copy(gradle copy)

>./gradlew tasks

>./gradlew properties

Gradle构建 DSL（Domain Specific Lauguage:领域定义语言）

1.脚本的代理对象

| Type of script | Delegates to instance of |
| --- | --- |
| Build script | [Project](https://docs.gradle.org/current/dsl/org.gradle.api.Project.html) |
| Init script | [Gradle](https://docs.gradle.org/current/dsl/org.gradle.api.invocation.Gradle.html) |
| Settings script | [Settings](https://docs.gradle.org/current/dsl/org.gradle.api.initialization.Settings.html) |

2.脚本块

| Block | Description |
| --- | --- |
| [allprojects { }](https://docs.gradle.org/current/dsl/org.gradle.api.Project.html" \l "org.gradle.api.Project:allprojects(groovy.lang.Closure)) | 配置项目和子项目 |
| [artifacts { }](https://docs.gradle.org/current/dsl/org.gradle.api.Project.html" \l "org.gradle.api.Project:artifacts(groovy.lang.Closure)) | 配置项目的发行物 |
| [buildscript { }](https://docs.gradle.org/current/dsl/org.gradle.api.Project.html" \l "org.gradle.api.Project:buildscript(groovy.lang.Closure)) | 配置项目构建脚步的classpath |
| [configurations { }](https://docs.gradle.org/current/dsl/org.gradle.api.Project.html" \l "org.gradle.api.Project:configurations(groovy.lang.Closure)) | 配置项目的依赖配置 |
| [dependencies { }](https://docs.gradle.org/current/dsl/org.gradle.api.Project.html" \l "org.gradle.api.Project:dependencies(groovy.lang.Closure)) | 配置依赖 |
| [repositories { }](https://docs.gradle.org/current/dsl/org.gradle.api.Project.html" \l "org.gradle.api.Project:repositories(groovy.lang.Closure)) | 配置以来仓库 |
| [sourceSets { }](https://docs.gradle.org/current/dsl/org.gradle.api.Project.html" \l "org.gradle.api.Project:sourceSets(groovy.lang.Closure)) | 配置项目的源集 |
| [subprojects { }](https://docs.gradle.org/current/dsl/org.gradle.api.Project.html" \l "org.gradle.api.Project:subprojects(groovy.lang.Closure)) | 配置当前项目的子项目 |
| [publishing { }](https://docs.gradle.org/current/dsl/org.gradle.api.Project.html" \l "org.gradle.api.Project:publishing(groovy.lang.Closure)) | 配置发行插件的 [PublishingExtension](https://docs.gradle.org/current/dsl/org.gradle.api.publish.PublishingExtension.html) |

构建**Android APP**

**1.**根层级的**build.gradle**文件

// Top-level build file where you can add configuration options common to all sub-projects/modules.

buildscript {

repositories {...............................1

google()

jcenter()

}

dependencies {.............................2

classpath 'com.android.tools.build:gradle:3.0.1'

// NOTE: Do not place your application dependencies here; they belong

// in the individual module build.gradle files

}

}

allprojects {............................................3

repositories {

google()

jcenter()

}

}

task clean(type: Delete) {.......................4

delete rootProject.buildDir

}

|  |
| --- |
|  |

**1.**下载插件模块

**2.gradle android** 插件

**3.**根层级配置，模块配置

**4.clean** 任务

**2.app**模块的**build.gradle**

apply plugin: 'com.android.application'

android {

compileSdkVersion 26

defaultConfig {

applicationId "org.gradle.helloworldgradle"

minSdkVersion 19

targetSdkVersion 26

versionCode 1

versionName "1.0"

testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"

}

buildTypes {

release {

minifyEnabled false

proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'

}

}

}

**3.library**模块的**build.gradle**

apply plugin: 'com.android.library'

其他没有和app模块的build.gradle没有区别

4.运行Gradle任务

·命令行

>./gradle build

>./gradle assembleDebug

>./gradle assembleRelease

·使用Gradle窗口

