

HenCoder Plus 讲义

Gradle 配置文件拆解

gradle 是什么

- 是构建工具，不是语言
- 它用了 Groovy 这个语言，创造了一种 DSL，但它本身不是语言

怎么构建？

- 按照 gradle 的规则（build.gradle、settings.gradle、gradle-wrapper、gradle 语法）

闭包

- Java 的 Lambda 表达式：是单抽象方法（SAM）的接口的匿名类对象的快捷写法，只是一个语法糖。
- Kotlin 的 Lambda 表达式：和匿名函数相当，实质上是一个函数类型的对象，并不只是语法糖。
- Groovy 的 Lambda 表达式：Groovy 里不叫「Lambda」表达式，而是叫「闭包」；在功能上，和 Kotlin 的 Lambda 比较相似，都是一个「可以传递的代码块」，具体的功能比 Kotlin 的 Lambda 更强一些，但基本的概念是一样的。

为什么 Groovy 可以写出类似 JSON 格式的配置？

因为它们其实都是方法调用，只是用闭包来写成了看起来像是 JSON 型的格式。

buildTypes 和 productFlavors

这个.....还是看视频吧

compile, implementation 和 api

- implementation：不会传递依赖
- compile / api：会传递依赖；api 是 compile 的替代品，效果完全等同
- 当依赖被传递时，二级依赖的改动会导致 0 级项目重新编译；当依赖不传递时，二级依赖的改动不会导致 0 级项目重新编译

Gradle Wrapper

- 通过「只同步版本，不同步文件」的方式来减小协作项目的大小
- 每个人电脑上的 Gradle 存放在固定位置，然后使用 Gradle Wrapper 的配置来取用对应的版本就行了

task

- 使用方法： `./gradlew taskName`
- task 的结构：

```
task taskName {  
    初始化代码  
    doFirst {  
        task 代码  
    }  
    doLast {  
        task 代码  
    }  
}
```

- doFirst() doLast() 和普通代码段的区别：
 - 普通代码段：在 task 创建过程中就会被执行，发生在 configuration 阶段
 - doFirst() 和 doLast()：在 task 执行过程中被执行，发生在 execution 阶段。如果用户没有直接或间接执行 task，那么它的 doLast() doFirst() 代码不会被执行
 - doFirst() 和 doLast() 都是 task 代码，其中 doFirst() 是往队列的前面插入代码，doLast() 是往队列的后面插入代码

- task 的依赖：可以使用 `task taskA(dependsOn: b)` 的形式来指定依赖。指定依赖后，task 会在自己执行前先执行自己依赖的 task。

gradle 执行的生命周期

三个阶段：

- 初始化阶段：执行 settings.gradle，确定主 project 和子 project
- 定义阶段：执行每个 project 的 build.gradle，确定出所有 task 所组成的有向无环图
- 执行阶段：按照上一阶段所确定出的有向无环图来执行指定的 task

在阶段之间插入代码：

- 一二阶段之间：
 - settings.gradle 的最后
- 二三阶段之间：

```
afterEvaluate {  
    插入代码  
}
```

问题和建议？

课上技术相关的问题，都可以去群里和大家讨论，对于比较通用的、有价值的问题，可以去我们的知识星球提问。

具体技术之外的问题和建议，都可以找丢物线（微信：diuwuxian），丢丢会为你解答技术以外的一切。



觉得好？

如果你觉得课程很棒，欢迎给我们好评呀！<https://ke.qq.com/comment/index.html?cid=381952>

一定要是你真的觉得好，再给我们好评。不要仅仅因为对扔物线的支持而好评（报名课程已经是你最大的支持了，再不够的话 B 站多来点三连我也很开心），另外我们也坚决不做好评返现等任何的交易。我们只希望，在课程对你有帮助的前提下，可以看到你温暖的评价。

更多内容：

- 网站：<https://hencoder.com>；<https://kaixue.io>
- 各大搜索引擎、微信公众号、微博、知乎、掘金、哔哩哔哩、YouTube、西瓜视频、抖音、快手、微视：统一账号「扔物线」，我会持续输出优质的技术内容，欢迎大家关注。
- 哔哩哔哩快捷传送门：<https://space.bilibili.com/27559447>

大家如果喜欢我们的课程，还请去扔物线的哔哩哔哩，帮我素质三连，感谢大家！