

# HenCoder Plus 第 20 课 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

- 相当于可以被传递的代码块

### buildType 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

### 项目结构

- 单 project：build.gradle
- 多 project：由 settings.gradle 配置多个

查找 settings 的顺序：

1. 当前目录
2. 兄弟目录 master
3. 父目录

### 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
}

```