参考：<http://blog.csdn.net/pkaq_/article/details/53584540>

我们了解到一个典型的多模块项目需要有一个根模块项目以及模块描述文件(默认为settings.gradle)，下面我们将通过一个 [多项目示例](https://github.com/GradleCN/GradleSide/tree/master/17-multi-project-central)来对多模块项目的结构以及脚本进行更深入的了解。

如果要让 Gradle 支持多模块构建，只需为你的项目确定一个根模块项目 并且在根模块项目下添加 settings.gradle 文件用以描述模块（项目）关系

include 'base','main'

　　这里将模块（项目）名称路径的字符串以数组的形式传递给了include函数，Gradle 会以相对于当前目录按照 include 给定的模块（项目）路径查找对应的子模块，如果要声明的模块（项目）有多个层次可以用:进行描述，假设main下面又分了Alpha,Bravo 那么声明方式则按如下方式书写

include 'base','main:Alpha','main:Bravo'

　　这里的 settings 文件实际上是对应的 Settings 接口，Setitings 接口中提供的函数在本脚本中都是可用的，具体细节请查阅 [>Settins接口DSL<](https://docs.gradle.org/current/dsl/) 文档进行了解。

　　在构建的 **初始化阶段**，Gradle 会读取这个文件，并创建一个 Settings 类型的实例。Gradle 会依据此文件进行多模块项目构建，默认情况下 Gradle 会从同级的 master 目录下寻找此文件，如果未找到则会去父级目录寻找。如果搜寻不到 settings 文件，那么 Gradle 会把模块当做单独构建的项目去执行单独构建。这里 Gradle 提供了一个命令行参数 -u 或 --no-search-upward 来控制 Gradle 不去父目录搜寻 settings 文件。

自定义初始化阶段逻辑

　　上面已经提到由于 Settings 文件会在 **初始化阶段** 读取执行，所以我们也可以将一些需要在初始化阶段执行的逻辑放到此处，如下面的示例

include 'base','main' println

def initFunc(){

println "<!-- three little birds -->" }

initFunc()

执行Gradle build会发现在初始化阶段已经分别输出了两行语句。

< ---- build init ---- >

<!-- three little birds -->

自定义setting文件名

　　默认情况下，Gradle 会采用settings.gradle作为文件名去查找，但如果处于某种不可描述的原因要采用其它名称的话也是可以的，调用命令时可以通过-c或者--settings-file参数来指定settings文件的位置以及名称。

tip: 通常情况下，不存在孤立的子模块，所以构建的执行顺序往往是由依赖顺序所决定的。但如果无法通过依赖顺序决定，那么 Gradle 会简单的按照首字母顺序决定构建顺序。