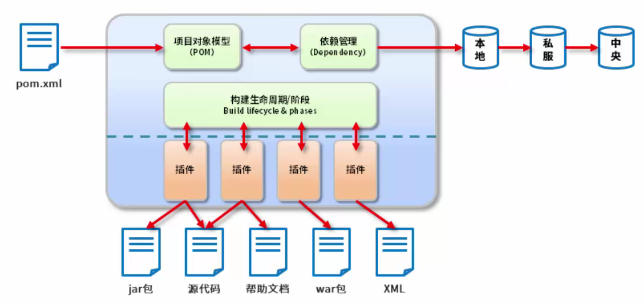
# Maven 基础

#### 1.Maven是什么

本质：一个项目管理工具，将项目开发和管理过程抽象成一个项目对象模型（POM）

POM(Project Object Model)：项目对象模型



Maven指的是蓝色包含的部分

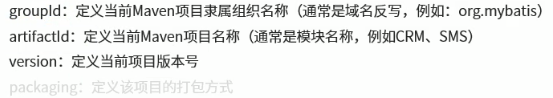
作用：项目构建、依赖管理、统一开发结构

#### 2.Maven基础概念

1. 仓库：用于存储资源，包含各种jar包



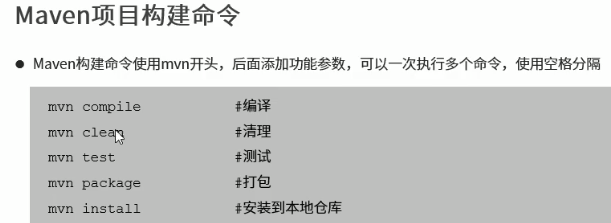
2. 坐标：定位仓库中资源的位置

主要组成：

一定要记住的网站：mvnrepository.com 去查询资源的坐标。（粘贴到pom.xml中）

作用：唯一标识，唯一性定位资源位置，可以将资源的识别与下载工作交由机器完成。

3.项目的构建



插件创建工程



#### 3.Maven依赖管理

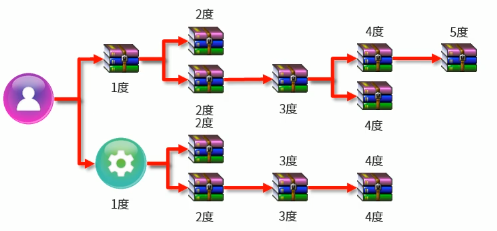
1. 依赖配置



2.依赖传递

直接依赖：当前项目中通过依赖配置建立的依赖关系

间接依赖：被资源的资源如果依赖其他资源，当前项目间接依赖其他资源

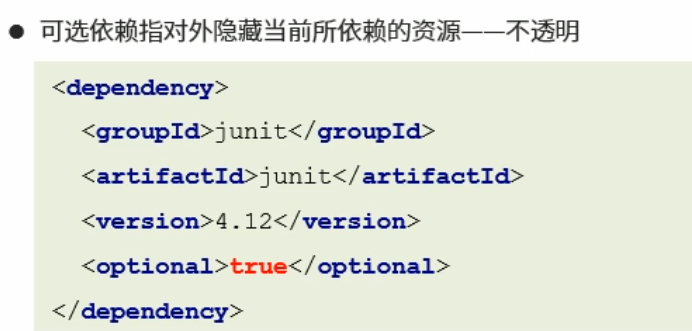


依赖冲突：1度覆盖2度，2度覆盖3度…

声明优先：度相同谁先配置谁生效

特殊优先：同级配置了相同资源的不同版本，后配置的覆盖先配置的。

可选依赖与排除依赖：



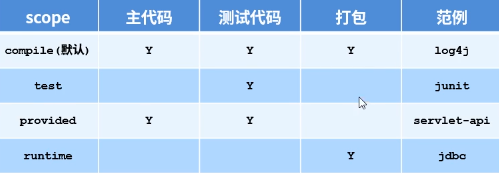


3.依赖范围

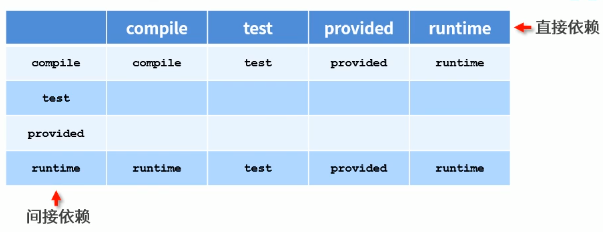
主程序范围(main文件夹范围内)

测试程序范围有效（test文件夹范围内）

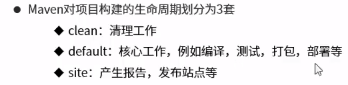
是否参与打包（package指令范围内）

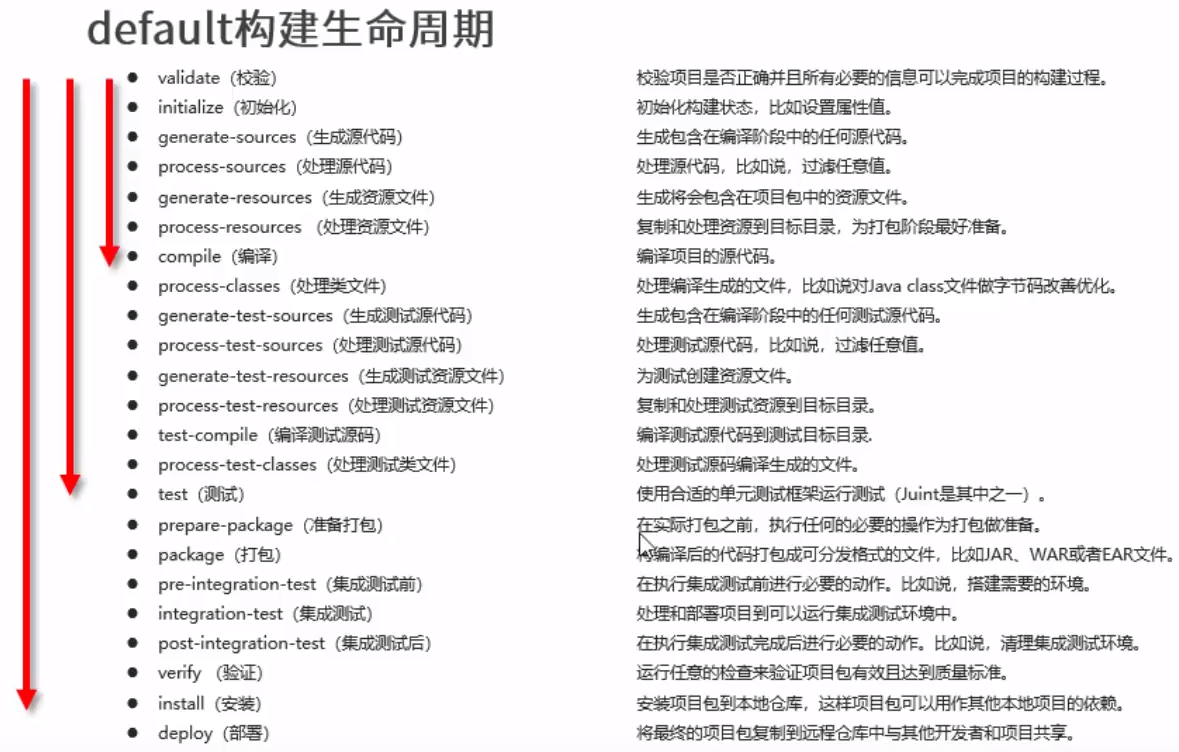


依赖范围的传递性



#### 4. 生命周期与插件



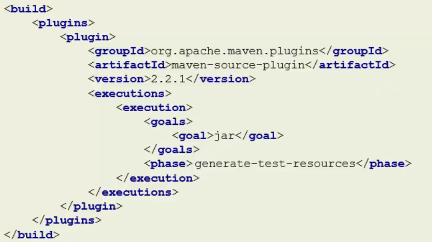


2.插件

插件与生命周期内的阶段绑定，执行到对应生命周期时执行对应插件功能

默认maven在各个生命周期上绑定有预设的功能

通过插件可以自定义其他功能



executions execution 标记插件执行在什么时候。

phase执行时间 对应上上表中的时间

goals goal 执行目标

# Maven 高级（学完svm再来）

#### 1.分模块开发与设计

#### 2.聚合

#### 3.继承

#### 4.属性

#### 5.版本管理

#### 6.资源配置

#### 7.多环境开发

#### 8.跳过测试

#### 9.私服