1. Maven基础
   1. POM：工程对象模型
   2. Pom文件
      1. 每个工程都只有一个pom.xml文件
      2. 无论是否显示定义，每个pom.xml文件都有一个父类pom.xml文件
      3. Pom.xml文件的根节点是project，它必须含有以下三个元素
         1. GrouoID： 项目标识
         2. ArtifactId：模块标识
         3. Version ： 版本号
   3. Meavn生命周期
      1. 定义：maven命令从运行到结束都在声明周期里操作（不是所有）
      2. 抽象：maven的生命周期是抽象的，它本身并不做任何的工作。实际的工作都交由"插件"来完成。
      3. 分类
         1. Clean：在进行真正的构建之前进行一些清理工作
            1. 组成阶段

Pre-clean

Clean

Post-clean

* + - 1. Default：构建的核心部分，编译，测试，打包，部署等等
         1. 重要组成阶段（部分）

Compile

Test

Package

Install

Deploy

* + - 1. Site：生成项目报告，站点，发布站点
         1. 组成阶段

Pre-site

Site

Post-site

Site-deploy

* + 1. 命令运行过程举例
       1. Mvn clean
          1. Pre-clean -> clean
       2. Package
          1. Compile->Test’->Package
       3. 结论
          1. 由此可见，执行一个maven命令时，会把其生命周期前所有的过程都执行。但不要误认为clean、default、site生命周期有任何关联。这三个是完全独立的生命周期。
  1. Maven仓库
     1. 分类
        1. 本地仓库
           1. 在当前用户目录下的.m2文件夹下
        2. 远程仓库
           1. 私服

一般是公司内部的仓库服务器

* + - * 1. 中央仓库

Maven自己的仓库

服务器在国外下载速度慢

阿里云CSDN的镜像仓库

服务器在国内，速度快

* + 1. 取所需文件的过程
       1. 本地->私服->中央仓库
  1. Maven命令
     1. 分类
        1. 生命周期命令
           1. 是插件命令的集合
        2. 插件命令
           1. 有些插件目标可以绑定生命周期的某个阶段，而有插件目标不适合绑定到生命周期中。
        3. 区别
           1. 插件命令只是生命周期中的一小部分工作，适合用来做单元性的工作。
           2. 生命周期命令适合用来做构键工作
     2. 格式
        1. 插件命令
           1. Mvn+插件别名+目标名
           2. Mvn+项目名+模块名+版本+目标
        2. 生命周期命令
           1. Mvn+生命周期名
     3. 目标
        1. **定义：**表示一个特定的、对构建和管理工程有帮助的任务。它可能绑定了 0 个或多个构建阶段。没有绑定任何构建阶段的目标可以在构建生命周期之外被直接调用执行。
  2. 快照
     1. 每次构件项目时都会从远程仓库获取最新的快照版本（假设你的工作每天都要提交，但是不是快照版本的话，maven又不会自动去帮你下载最新的开发包）
  3. 常用命令
     1. Archetype：generate
     2. Eclipse：eclipse
     3. Mvn help：describe –Dplugins：xxx