mvn help:system

该命令会打印所有的java系统属性和环境变量

配置文件 proxys标签设置代理

maven根目录下conf中的settings.xml配置全局，~m2/settings.xml配置用户的，但是用户自定义配置优先级高与全局配置

ide最好不要使用内嵌的maven，如果使用了的话可能导致和命令行构件行为的不一致

modelVersion指定的是pom模型版本，与maven的版本不同

基本坐标：groupId artifactId version

snapshot 快照说明该项目还处在开发中，是不稳定版本

可以定义一个name来设置一个名称

测试代码只在运行测试时用到，不会被打包，项目的主代码会被打包到最终的构件中国

mvn clean compile

编译项目

clean命令告诉maven清理输出目录target/，compile命令告诉maven编译主项目代码

dependencies 下可以包含多个dependency 元素

scope表示依赖范围，如果为test则表示只对测试有效，默认值为compile，表示依赖对主代码和测试代码都有效

maven 3规定测试方法都要以test开头

mvn clean test

调用maven执行测试

maven clean package

将项目打包

maven会在打包之前执行编译测试等操作

mvn clean install

安装任务，将项目输出jar安装到本地仓库中

mvn archetype:generate

引导创建一个maven项目

依赖管理的底层基础：坐标

maven坐标的基本元素包括groupId artifactId version packaging classifier

packaging元素定义maven项目的打包方式，默认值为jar

classifier 元素帮助定义构建输出一些附属构件

maven在编译主代码的时候需要使用一套classpath，在编译和执行测试的时候会使用另一套classpath，实际运行maven项目的时候又会使用一套classpath

依赖范围就是用来控制依赖与这3种classpath（编译classpath 测试classpath 和运行classpath）的关系

scope值：

1. compile 编译依赖范围，默认值，对于编译 测试 和运行classpath 都有效

2. test 测试依赖范围，只对测试classpath有效

3. provided 已提供依赖范围，对于编译和测试classpath有效，但在运行时无效

4. runtime 运行时依赖范围，对于测试和运行classpath有效，但在编译主代码时无效

5. system 系统依赖范围，与provided一样，但是需要通过systemPath指定依赖文件的绝对路径

6. import 导入依赖范围，该依赖不会对3种classpath产生实际的影响

传递性依赖的范围

maven调解的第一原则是 路径最近者优先，第二原则是第一声明者优先

可选依赖：不会传递依赖，需要显示声明依赖

排除依赖：

<dependency>  
<groupId>com.juvenxu.mvnbook</groupId>  
<artifactId>project-b</artifactId>  
<version>1.0.0</version>  
<exclusions>  
<exclusion>  
<groupId>com.juvenxu.mvnbook</groupId>  
<artifactId>project-c</artifactId>  
</exclusion>  
</exclusions>  
</dependency>  
<dependency>  
<groupId>com.juvenxu.mvnbook</groupId>  
<artifactId>project-c</artifactId>  
<version>1.1.0</version>  
</dependency>  
</dependencies>

## 归类依赖

<properties>  
<springframework.version>2.5.6</springframework.version> 定义一个版本  
</properties>  
<dependencies>  
<dependency>  
<groupId>org.springframework</groupId>  
<artifactId>spring-core</artifactId>  
<version>${springframework.version}</version> 引用定义的版本  
</dependency>  
<dependency>  
<groupId>org.springframework</groupId>  
<artifactId>spring-beans</artifactId>  
<version>${springframework.version}</version>  
</dependency>

## 优化依赖

mvn dependency:list 可以查看当前项目的已解析依赖，包括直接依赖和传递性依赖

mvn dependency:tree 查看依赖树

显示声明任何项目中用到的依赖

一个 WAR 包下至少包含两个子目录：META-INF 和 WEB-INF，前者包含了一些  
打包元数据信息，我们一般不去关心；后者是 WAR 包的核心，WEB-INF 下必须  
包含一个 Web 资源表述文件 web.xml，它的子目录 classes 包含所有该 Web 项目的类，而另一个子目录 lib 则包含所有该 Web 项目的依赖 JAR 包，classes 和lib 目录都会在运行的时候被加入到 Classpath 中

Web 项目比较特殊的地方在于：它还有一个 Web 资源目录，其默认位置是 src/main/webapp/