

阶段	处理	描述
验证 validate	验证项目	验证项目是否正确且所有必须信息是可用的
编译 compile	执行编译	源代码编译在此阶段完成
测试 Test	测试	使用适当的单元测试框架（例如JUnit）运行测试。
包装 package	打包	创建JAR/WAR包如在 pom.xml 中定义提及的包
检查 verify	检查	对集成测试的结果进行检查，以保证质量达标
安装 install	安装	安装打包的项目到本地仓库，以供其他项目使用
部署 deploy	部署	拷贝最终的工程包到远程仓库中，以共享给其他开发人员和工程

三个标准的生命周期：

- **clean**：项目清理的处理
- **default(或 build)**：项目部署的处理
- **site**：项目站点文档创建的处理

构建阶段由插件目标构成

一个插件目标代表一个特定的任务（比构建阶段更为精细）。例如：
clean 和 package 是构建阶段，dependency:copy-dependencies 是目标。

```
mvn clean dependency:copy-dependencies package
```

这里的 clean 阶段将会被首先执行，然后 dependency:copy-dependencies 目标会被执行，最终 package 阶段被执行。

Clean 生命周期

当我们执行 mvn post-clean 命令时，Maven 调用 clean 生命周期，它包含以下阶段：

- pre-clean : 执行一些需要在clean之前完成的工作
- clean : 移除所有上一次构建生成的文件
- post-clean : 执行一些需要在clean之后立刻完成的工作

mvn clean 中的 clean 就是上面的 clean，在一个生命周期中，运行某个阶段的时候，它之前的所有阶段都会被运行。

Default (Build) 生命周期

生命周期阶段	描述
validate	检查工程配置是否正确，完成构建过程的获取到。
initialize	初始化构建状态，例如设置属性。
generate-sources	生成编译阶段需要包含的任何源码文件。
process-sources	处理源代码，例如，过滤任何值（filter）。
generate-resources	生成工程包中需要包含的资源文件。
process-resources	拷贝和处理资源文件到目的目录中，为打包做准备。
compile	编译工程源码。
process-classes	处理编译生成的文件，例如 Java Class 文件。
generate-test-sources	生成编译阶段需要包含的任何测试源代码。
process-test-sources	处理测试源代码，例如，过滤任何值（filter）。
test-compile	编译测试源代码到测试目的目录。
process-test-classes	处理测试代码文件编译后生成的文件。
test	使用适当的单元测试框架（例如JUnit）进行测试。
prepare-package	在真正打包之前，为准备打包执行任何必要的操作。
package	获取编译后的代码，并按照可发布的格式生成 WAR 或者 EAR 文件。
pre-integration-test	在集成测试执行之前，执行所需的操作。设置环境变量。
integration-test	处理和部署必须的工程包到集成测试能够运行的环境。

post-integration-test	在集成测试被执行后执行必要的操作。例
verify	运行检查操作来验证工程包是有效的，并
install	安装工程包到本地仓库中，该仓库可以作 赖。
deploy	拷贝最终的工程包到远程仓库中，以共享 程。

Site 生命周期

Maven Site 插件一般用来创建新的报告文档、部署站点等。

- pre-site：执行一些需要在生成站点文档之前完成的工作
- site：生成项目的站点文档
- post-site：执行一些需要在生成站点文档之后完成的工作，并且为部署做准备
- site-deploy：将生成的站点文档部署到特定的服务器上（发布）