# Maven的生命周期

1. <https://www.runoob.com/maven/maven-build-life-cycle.html>

## 前言：

Maven有三套相互独立的生命周期，分别是：

1）Clean生命周期；

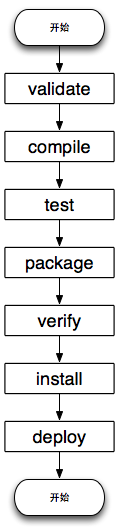
2）Default生命周期；

3）Site生命周期；

**每套生命周期都由一组阶段（Phase）组成**，平时我们在命令行输入的命令总会对应于一个特定的阶段。

## 零、典型的构建周期：

一个典型的Maven构建周期由以下几个阶段的序列组成：



|  |  |  |
| --- | --- | --- |
| 阶段 | 处理 | 描述 |
| validate | 验证项目 | 验证项目是否正确且所有必须信息是可用的 |
| compile | 执行编译 | 源代码编译在此阶段完成 |
| test | 测试 | 使用适当的单元测试框架进行测试 |
| package | 打包 | 创建JAR/WAR包 |
| verify | 检查 | 对集成测试的结果进行检查,以保证质量达标 |
| install | 安装 | 安装打包的项目到本地仓库,以供其他项目使用 |
| 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 |

## 一、Clean生命周期：

在进行真正的构建之前执行一些清理工作。

Clean生命周期一共包含三个阶段：

1）pre-clean：执行一些需要在clean之前完成的工作；

2）clean：移除所有上一次构建生成的文件；

3）post-clean：执行一些需要在clean之后立刻完成的工作；

比如，**mvn clean** 就是执行Clean生命周期的clean阶段。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，也就是说，**mvn clean** 等同于 **mvn pre-clean clean**

## 二、Site生命周期：

1）pre-site：执行一些需要在生成站点文档之前的工作；

2）site：生成项目的站点文档；

3）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；

4）site-deploy：将生成的站点文档部署到特定的服务器上；

## 三、Default生命周期：

这里解释一些比较重要和常用的阶段。

|  |  |
| --- | --- |
| 生命周期阶段 | 描述 |
| validate(校验) | 校验项目是否正确并且所有必要的信息可以完成项目的构建过程 |
| initialize(初始化) | 初始化构建状态,比如设置属性值 |
| generate-source  (生成源代码) | 生成包含在编译阶段中的任何源代码 |
| process-source  (处理源代码) | 处理源代码,比如:过滤任意值 |
| generate-resource  (生成资源文件) | 生成将会包含在项目中的资源文件 |
| process-resource  (处理资源文件) | 复制和处理资源到目标目录,为打包阶段做准备 |
| compile(编译) | 编译项目的源代码 |
| process-class  (处理类文件) | 处理编译生成的文件,比如说对Java Class文件做字节码改善优化 |
| generate-test-source  (生成测试源代码) | 生成包含在编译阶段中的任何测试源代码 |
| process-test-source  (处理测试源代码) | 处理测试源代码,比如说,过滤任意值 |
| generate-test-resource  (生成测试资源文件) | 为测试创建资源文件 |
| process-test-resource  (处理测试资源文件) | 复制和处理测试资源到目标目录 |
| test-compile  (编译测试源码) | 编译测试源码到测试目标目录 |
| process-test-classes  (处理测试资源文件) | 处理测试源代码生成的文件 |
| test(测试) | 使用合适的测试单元框架进行测试 |
| prepare-package  (准备打包) | 在实际打包之前,执行任何的必要的操作为打包做准备 |
| package(打包) | 将编译后的代码打包成可分发格式的文件,比如JAR、WAR或者EAR文件 |
| pre-integration-test  (集成测试前) | 在执行继承测试前进行必要的动作。 |
| integration-test  (集成测试) | 处理和部署项目到可以运行集成测试环境中 |
| post-integration-test  (集成测试后) | 在执行集成测试后进行必要的动作 |
| verify(验证) | 运行任意的检查来验证项目包有效且达到质量标准 |
| install(安装) | 安装项目到本地仓库,这样项目包可以用作其他本地项目的依赖 |
| deploy(部署) | 将最终的项目包复制到远程仓库中与其他开发者和项目共享 |