**Maven命令：**

* **创建一个简单的Java工程：**

mvn archetype:create

-DgroupId=com.mycompany.example

-DartifactId=Example

* **创建一个java的web工程：**

mvn archetype:create -DarchetypeGroupId=org.apache.maven.archetypes

-DarchetypeArtifactId=maven-archetype-webapp

-DgroupId=com.mycompany.app

-DartifactId=my-webapp

* 安装当前工程的输出文件到本地仓库: mvn install
* 安装指定文件到本地仓库：

mvn install:install-file

-DgroupId=<groupId>

-DartifactId=<artifactId>

-Dversion=1.0.0

-Dpackaging=jar

-Dfile=<myfile.jar>

* 上传到中央仓库：mvn deploy

1. 只有JAR/WAR包，但没有POM文件

|  |
| --- |
| mvn deploy:deploy-file \      -DgroupId=com.somecompany \      -DartifactId=project \      -Dversion=1.0.0 \      -DgeneratePom=true \      -Dpackaging=jar \      -DrepositoryId=releases \      -Durl=http://maven.dtdream.com/content/repositories/releases/ \      -Dfile=target/project-1.0.0.jar |

2.有JAR/WAR包，也有pom文件

|  |
| --- |
| mvn deploy:deploy-file \      -DgeneratePom=false \      -DrepositoryId=releases \      -Durl=http://maven.dtdream.com/content/repositories/releases/ \      -DpomFile=pom.xml \          -Dfile=target/project-1.0.0.jar |

* 打包：mvn package
* 编译：mvn compile
* 编译测试程序：mvn test-compile
* 清空：mvn clean
* 运行测试：mvn test
* 分析项目的依赖信息：mvn dependency:analyze 或 mvn dependency:tree
* 生成eclipse项目文件: mvn eclipse:eclipse
* 查看帮助信息：mvn help:help 或 mvn help:help -Ddetail=true
* 查看插件的帮助信息：mvn <plug-in>:help，比如：mvn dependency:help 或 mvn ant:help 等等。