Refer : <https://maven.apache.org/guides/getting-started/maven-in-five-minutes.html>

* To generate maven project in non interactive mode

mvn archetype:generate -DgroupId=com.psl.app -DartifactId=first-app -DarchetypeArtifactId=maven-archetype-quickstart -DarchetypeVersion=1.4 -DinteractiveMode=false

cd first-app/ ~ whear pom folder resides

mvn compile

mvn test

mvn package

mvn install

java -cp target/my-app-1.0-SNAPSHOT.jar com.psl.app.App

------------------------------------------------------------------------------------------------------------------------------------------

* To generate maven project in interactive mode

mvn archetype:generate

* Then enter the no. of archtype you want to generate from list of 2935 archtypes

for example : quickstart archetype has no. : 1785

* Then it will ask you for the version,their will be a list of version which are associated with particular version ,so type the no. you want a version of

ex: 8

8 is associated with (version 1.4)

* Then it will ask for group Id (it’s unique id , sometimes package name is same as group id, but a project can have multiple packages but only one group id)

Ex: com.comapanyname.applicationname

* Then artifact id

Ex:MyFirstApp (name you want to give to your application)

* Then version

Ex: 1.0-Snapshot

* Then package name

Ex: com.company.app (can be same as group id and can also be different)

------------------------------------------------------------------------------------------------------------------------------------------

Tutorial : <https://www.tutorialspoint.com/maven/index.htm>

Maven is a build automation tool primarily used for Java projects. It helps to manage the build process, dependencies, and documentation for a project. Maven uses a declarative approach to build projects, where developers specify the project's dependencies, goals, and configuration in a file called a POM (Project Object Model).

Maven provides a central repository for managing dependencies, which simplifies the process of including external libraries in a project. It also provides a set of predefined build lifecycle phases, such as compile, test, package, and install, that developers can use to build their projects. Maven plugins can extend the functionality of Maven and provide additional features, such as generating documentation or running static code analysis tools.

Maven is widely used in the Java development community, and it can be integrated with many popular IDEs, such as Eclipse and IntelliJ IDEA. Its declarative approach and central repository make it easier to manage dependencies and build Java projects compared to traditional build tools.