Maven At A Glance

* **POM** – Project Object Model
* **mvn compile** – It compiles the source.java files into .class files and create a new target folder
* **maven-compiler-plugin** – It compiles the .java files
* **mvn test-compile** – it compiles test .java files
* **mvn clean** – to delete target directory
* **mvn test** – Used to run Junit test , It will compile source file , test file and then run junit test cases
* **mvn package** – package into jar file into target folder , it will also run Junit test cases from test folder

**execute the jar** – java –cp \*\*\*/\*\*\*.jar .class

* **Build Life Cycle** –
  + **Validate –** Validate POM in order , folder structure
  + **Compile –** Compiles .java and compile them in .class
  + **Test –** Execute Junit test cases
  + **Package –** Package all the class files
  + **Integration Test**
  + **Verify**
  + **Install –** Install the package into local repository
  + **Deploy -**
* **Gropu ID –** Package name
* **Artifact ID –** Project name
* **How to exclude transitive dependency**

<exclusions>

<exclusion>

<groupId></groupId>

<artifactId></artifactId>

</exclusion>

</exclusions>

* **Version Range in Maven POM**

**<>[4.1,] – Any version >4.1**

**<>[,4.1] – Any version less than or equal to 4.1**

**<>[,4.1) – do not include 4.1**

**<>[4.1,4.4] – highest one which is available**

* **<scope></scope**

Defines where jar should be used , at compile or Junit time

* **mvn help-effectivePom**
* **mvn archetype:generate –** will generate default project , type of project that we need