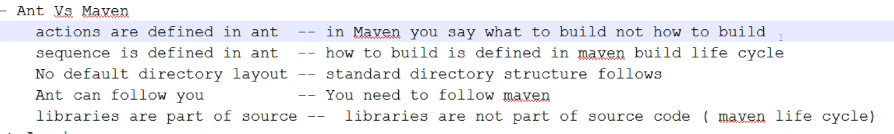
**Maven**

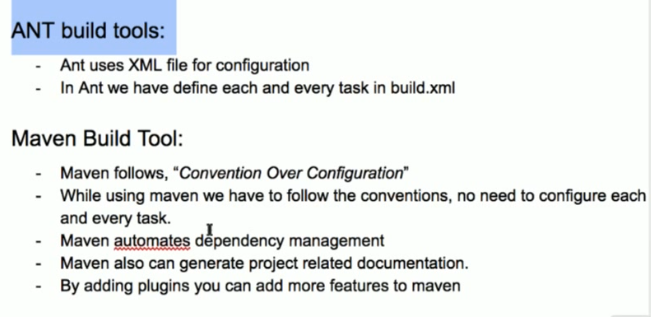
* Maven is the most widely used tool for java
* Gradle used for android
* MSBuild for .net
* Maven supports standardised build process
* Maven is also dependency manager tool
* Instead of using javac, java. We use maven
* Maven is a free open source tool by apache
* If we create a pom.xml, we can use the same in windows or Linux also

**Issues on ANT:**

* Not having standard directory structure
* So much of scripting
* Libraries are part of source code
* Code base size is kept growing and difficult to maintain in VCS
* There is no sequence of execution defined in ANT. Everything we need to define the sequence
* Maintainability & reliability of ant build script is difficult
* Ant is not following standards but automating it

**ANT vs Maven:**





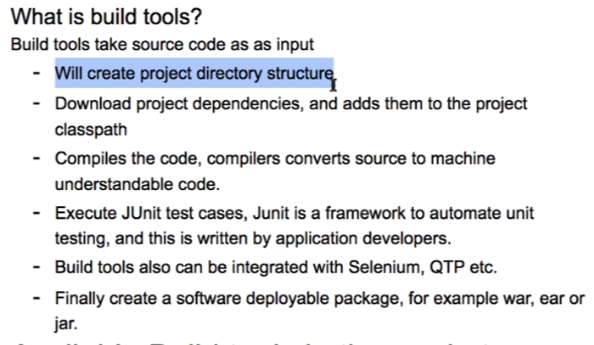
* Maven is from apache, we must install JDK for this
* **Mvn -version (to check the version)**

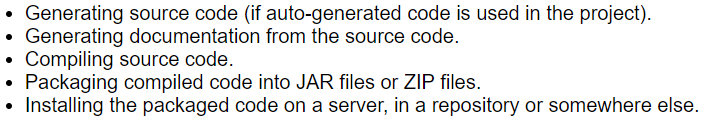
**Or**

* **Mvn -v**
* The script is in xml format. The default one is POM.xml (Project Object Model)

**Maven:**

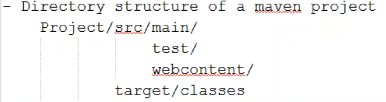
* Maven is built automation tool, project automation tool
* Build includes compile the code, unit testing, integration test, packaging the compiled code into jar files, bundling these files into a we archive or war files, deploying these files into server



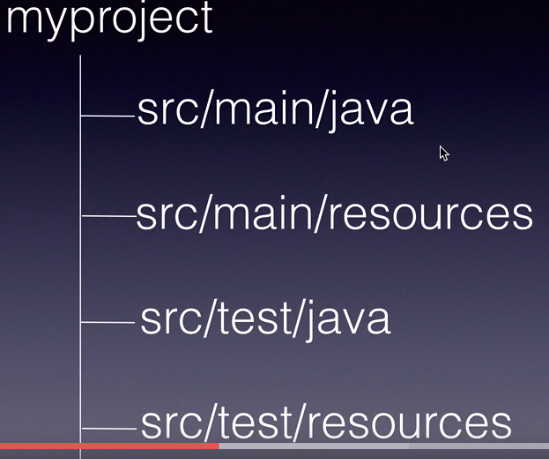




**Directory structure of maven project:**



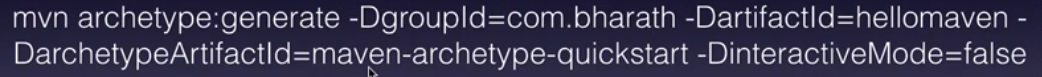
* Source code should be inside src. If we have test program, it should be inside test, Web content should be inside web content
* Everything will be inside src folder
* Once the build is done, the binaries will be created in the target folder
* We should follow this structure



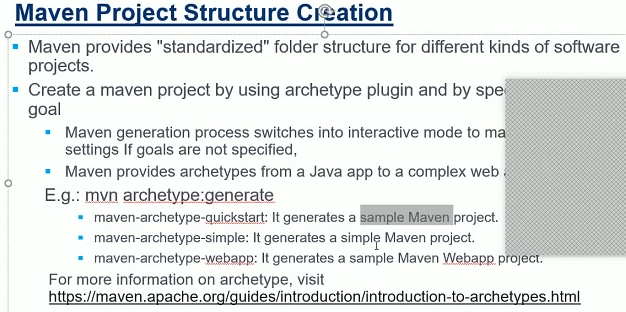
* Inside src, main contains main code and test contains test data inside src just like the main
* When we run the below command, it will download the libraries
* Source code should go into java folder
* Property files and xml files should go into resources folder
* Unit testing, integration testing should go into /test/java folder
* All property files, xml files that we use in test should go into /test/resources
* we no need to create all these folders, maven provides archetypes. These are like templates. Once we run that, it will create the directories for the application
* we can create different types of projects within IDE

**Generate a maven project:**

* The generate goal with archetype plugin and the install goal from the install plugin



* If we set the interactive mode as false, then the job runs in batch mode which takes the default values
* If we set it as true, then it will run in interactive mode which will ask us for the values
* If we don’t mention “DarchetypeArtifactId” then by default it will take “maven-archetype-quickstart”
* Archetype is the pluginid and generate is the goalid
* Maven uses the plugins to get the job done
* We can define all those in pom.xml





* As per above image, we are generating a project in QuickStart model
* There are many types of project creation
* Archetype generates the maven template project

**or**

* Run the below command for that. After running the below cmd, it will ask for the archetype id and groupid etc
* **Mvn archetype:generate**