1) What is Maven?

Maven is a project management tool. It is based on POM (Project Object Model). [More details.](https://www.javatpoint.com/maven-tutorial)

2) What aspects are managed by Maven?

Builds

Documentation

Reporting

SCMs

Releases

3) What are the advantages of Maven?

No need to add jar file in each project

Creates right directory structure

Builds and deploys the project

4) What is the command to check the maven version?

Type the following command on console to know the maven version.

mvn -version

5) What does the build tool?

Generates source code (if the auto-generated code is used)

Generates documentation from source code

Compiles source code

Packages compiled code into a JAR or ZIP file

Installs the packaged code in the local repository, server repository, or central repository

6) What is the difference between Ant and Maven?

|  |  |
| --- | --- |
| Ant | Maven |
| It is a toolbox. | It is a framework. |
| It is mainly a build tool. | It is mainly a project management tool. |
| There is no life cycle. | There is alife cycle. |
| Ant doesn't have formal conventions. | Maven has a convention to place source code, compiled code etc. |
| Ant is procedural. | Maven is declarative. |
| The ant scripts are not reusable. | The Maven plugins are reusable. |

7) Why is the use of the profile required in Maven?

For providing probability to projects, we use profiles.

PauseNext

Mute

Current TimeÂ 0:00

8) What is the syntax for offline project creation?

The syntax for project creation is:

mvn o packg.

9) How is the propagation of plugins to child POMs stopped?

It can be done using the following syntax:

set**<inherited>** to false.

10) What is the use of the exclusion element?

The element is used to exclude dependencies.

11) Define SNAPSHOT in terms of maven.

The snapshot indicates the current development copy.

12) Define Archetype.

It is a Maven plugin which is designed for the creation of project structure.

7. What does central repository consist of?

It consists of a large number of libraries that are frequently used.

13) Give the command for installation of the JAR file in a local repository.

mvn install

14) Mention the phases of cleaning lifecycle.

The lifecycle of cleaning consist of:

pre-clean

clean

post-clean

15) What is the purpose of mvn clean command?

The command removes the target directory before the starting of a build process.

16) What is a MOJO?

A MOJO stands for Maven plain Old Java Object. Each MOJO is an executable goal in Maven, and a plugin is a distribution of one or more related MOJOs.

17) What is a repository?

A repository is a directory or place where all the jars and pom.xml file are stored. There are 3 types of a repository in Maven:

Local Repository

Central Repository

Remote Repository

18) What is a local repository?

Maven local repository is created by maven in your local system when you run any maven command.

19) What is a central repository?

Maven community creates maven central repository on the web.

20) What is a remote repository?

Maven remote repository is located on the web by different vendors. So you need to define the dependency in pom.xml file manually. It is important because most of the libraries are missing from the central repository.

21) What is POM?

POM stands for Project Object Model. The pom.xml file contains information of project and project configuration

22) What are the build phases in Maven?

validate

compile

test

package

integration-test

verify

install

deploy

23) What is the command to package maven project?

mvn -package

24) What is the fully qualified artifact name of maven project?

**<groupId>**:**<artifactId>**:**<version>**

25) What is an archetype?

Archetype is the maven plugin. It creates the project structure.

MAVEN:

At first glance Maven can appear to be many things, but in a nutshell Maven is an attempt to apply patterns to a project's build infrastructure in order to promote comprehension and productivity by providing a clear path in the use of best practices. Maven is essentially a project management and comprehension tool and as such provides a way to help with managing:

Builds

Documentation

Reporting

Dependencies

SCMs

Releases

Distribution

How does Maven manage dependencies and transitive dependencies? Ans: Maven manages dependencies and transitive dependencies through the use of a Project Object Model (POM) file. This file contains information about the project, including a list of dependencies and their versions. When a project depends on another project, Maven will automatically include that project's dependencies (transitive dependencies) in the final build, eliminating the need for manual management of these dependencies. Maven uses a repository to store these dependencies, and it will automatically download them from the repository when they are needed. Additionally, Maven uses a dependency hierarchy to resolve conflicts between different versions of the same dependency, using the "nearest definition" strategy.

How to install Maven on a Linux machine?

Ans: [Click Here for Answer with code & Screenshots](https://draft.blogger.com/blog/post/edit/2695082220643154739/4650414199546157401)

What is a SNAPSHOT version?

     Ans:

Notice the value of the version tag in the pom.xml file shown below has the suffix: -SNAPSHOT.

<project xmlns="http://maven.apache.org/POM/4.0.0"

 ...

 <groupId>...</groupId>

 <artifactId>my-app</artifactId>

 ...

 <version>1.0-SNAPSHOT</version>

 <name>Maven Quick Start Archetype</name>

 ...

The SNAPSHOT value refers to the 'latest' code along with a development branch, and provides no guarantee the code is stable or unchanged. Conversely, the code in a 'release' version (any version value without the suffix SNAPSHOT) is unchanging.

In other words, a SNAPSHOT version is the 'development' version before the final 'release' version. The SNAPSHOT is "older" than its release.

During the [release](https://draft.blogger.com/blog/post/edit/2695082220643154739/4650414199546157401) process, a version of x.y-SNAPSHOT changes to x.y.

The release process also increments the development version to x.(y+1)-SNAPSHOT. For example, version 1.0-SNAPSHOT is released as version 1.0, and the new development version is version 1.1-SNAPSHOT.

3.What is the purpose of mvn clean command?

     Ans:    This command removes the target directory before the starting of a build process.

4. How to integrate Maven pom.xml with Jenkins job?

[Click Here For Steps with Screensho](https://draft.blogger.com/blog/post/edit/2695082220643154739/4650414199546157401)t

5. List out the dependency scope in Maven?

   The various dependency scopes used in Maven are:

     • Compile: It is the default scope, and it indicates what dependency is available in the classpath of the project

    • Provided: It indicates that the dependency is provided by JDK or web server or container at runtime

    • Runtime: This says that the dependency is not needed for compilation but is required during   execution

    • Test: It says dependency is available only for the test compilation and execution phases

    • System: It indicates you have to provide the system path

    • Import: This indicates that the identified or specified POM should be replaced with the dependencies in that POM’s section

6.  In Maven, what are the two settings files called and what are their locations?

In Maven, the settings files are called settings.xml, and the two settings files are located at

• Maven installation directory: $M2\_Home/conf/settings.xml

• User’s home directory: ${ user.home }/ .m2 / settings.xml

7. Maven dependencies with pom.xml for Automation Framework design? How to add selenium, testng,cucumber dependency in Maven?

[Click Here For POM.xml](https://draft.blogger.com/blog/post/edit/2695082220643154739/4650414199546157401)

This has the entire list dependencies required to design a Selenium Automation Framework from scratch.

8. [Dependency vs plugin](https://draft.blogger.com/blog/post/edit/2695082220643154739/4650414199546157401)

Both plugins and dependencies are Jar files.

But the difference between them is, most of the work in maven is done using plugins; whereas dependency is just a Jar file which will be added to the classpath while executing the tasks.

So, we can say, plugin is a Jar file which executes the task, and dependency is a Jar which provides the class files to execute the task.

9. Group id vs artifact id

      groupId will identify your project uniquely across all projects, so we need to enforce a naming schema. You have to follow the package name rules, which means that it has to be at least as a domain name you control, and you can create as many subgroups as you want.

eg. org.apache.maven, org.apache.commons

      artifactId is the name of the jar without version. If you created it then you can choose whatever name you want with lowercase letters and no strange symbols. If it's a third party jar you have to take the name of the jar as it's distributed.

eg. maven, commons-math

10. What is the default location for your local repository?

Answer: ~ / M2 / repository.

11. Maven plugins?

      1. surefire plugin

      2. compiler plugin

      3. resource plugin

12. What is the default scope that the maven uses if we do not specify the scope element in pom.xml?

Answer: Compile

How do I create documentation?

To get you jump started with Maven's documentation system you can use the archetype mechanism to generate a site for your existing project using the following command:

mvn archetype:generate \

  -DarchetypeGroupId=org.apache.maven.archetypes \

  -DarchetypeArtifactId=maven-archetype-site \

  -DgroupId=com.mycompany.app \

  -DartifactId=my-app-site

13. What scope can be used to make certain dependencies available only while compiling and running tests?

Answer: Test

14. What is archetype?

Answer: We will use maven archetype to generate the maven folder structure based on the   inputs.

Ex: mvn archetype:generate.

What is Apache Maven, and why is it used?

Apache Maven is a build automation tool used for managing and building projects. It simplifies project configuration, dependencies, and builds.

What is a POM file in Maven?

POM (Project Object Model) is an XML file that contains project configuration details, including dependencies, plugins, and build instructions.

How do you create a new Maven project using the command line?

To create a new Maven project, use the following command:  
arduino

mvn archetype:generate -DgroupId=com.example -DartifactId=my-project -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false

What is a Maven repository, and how does it work?

A Maven repository is a directory where Maven stores project dependencies. There are local repositories on your machine and remote repositories on the internet.

How can you specify dependencies in a Maven POM file?

You can specify dependencies in the <dependencies> section of the POM file. For example:

<dependencies>

    <dependency>

        <groupId>org.apache.commons</groupId>

        <artifactId>commons-lang3</artifactId>

        <version>3.12.0</version>

    </dependency>

</dependencies>

What is a Maven plugin, and how is it used?

A Maven plugin is an extension that provides additional functionality to Maven. Plugins are configured in the POM file and executed during the build process.

How do you build a Maven project from the command line?

To build a Maven project, use the following command:

mvn clean install

Explain the Maven build phases and their order.

Maven build phases include validate, compile, test, package, verify, install, and deploy, in that order. Each phase builds upon the previous one.

How do you skip a specific Maven build phase?

To skip a specific build phase, use the -Dskip.phaseName or -Dmaven.test.skip=true option. For example:

mvn clean install -DskipTests

What is the purpose of the settings.xml file in Maven?

The settings.xml file is used to configure Maven settings, such as repository locations, profiles, and authentication credentials.

What is a Maven profile, and how do you activate it?

A Maven profile is a set of configuration settings that can be activated based on conditions. You can activate a profile using the -P option. For example:

mvn clean install -PprofileName

What is the purpose of the mvn clean command, and when should it be used?

The mvn clean command removes the target directory and any compiled artifacts. It's useful when you want to start fresh or clean up your project.

How can you create a custom Maven archetype?

To create a custom archetype, you can use the mvn archetype:create-from-project command and then package it as a new archetype project.

Explain the difference between the compile and test scopes for dependencies.

Dependencies with the compile scope are needed for compilation and runtime, while those with the test scope are only needed for testing.

What is the Maven Central Repository, and why is it important?

The Maven Central Repository is a global repository of open-source artifacts. It's essential for sharing and accessing commonly used libraries.

How do you handle dependency conflicts in Maven?

You can use the <exclusions> element to exclude specific transitive dependencies or use the <dependencyManagement> section to specify preferred versions.

What is the purpose of the maven-surefire-plugin in Maven?

The maven-surefire-plugin is used for executing unit tests. It automatically runs test classes whose names match the \*Test pattern.

How do you generate a Maven project's site documentation?

To generate site documentation, use the mvn site command. It generates HTML-based documentation in the target/site directory.

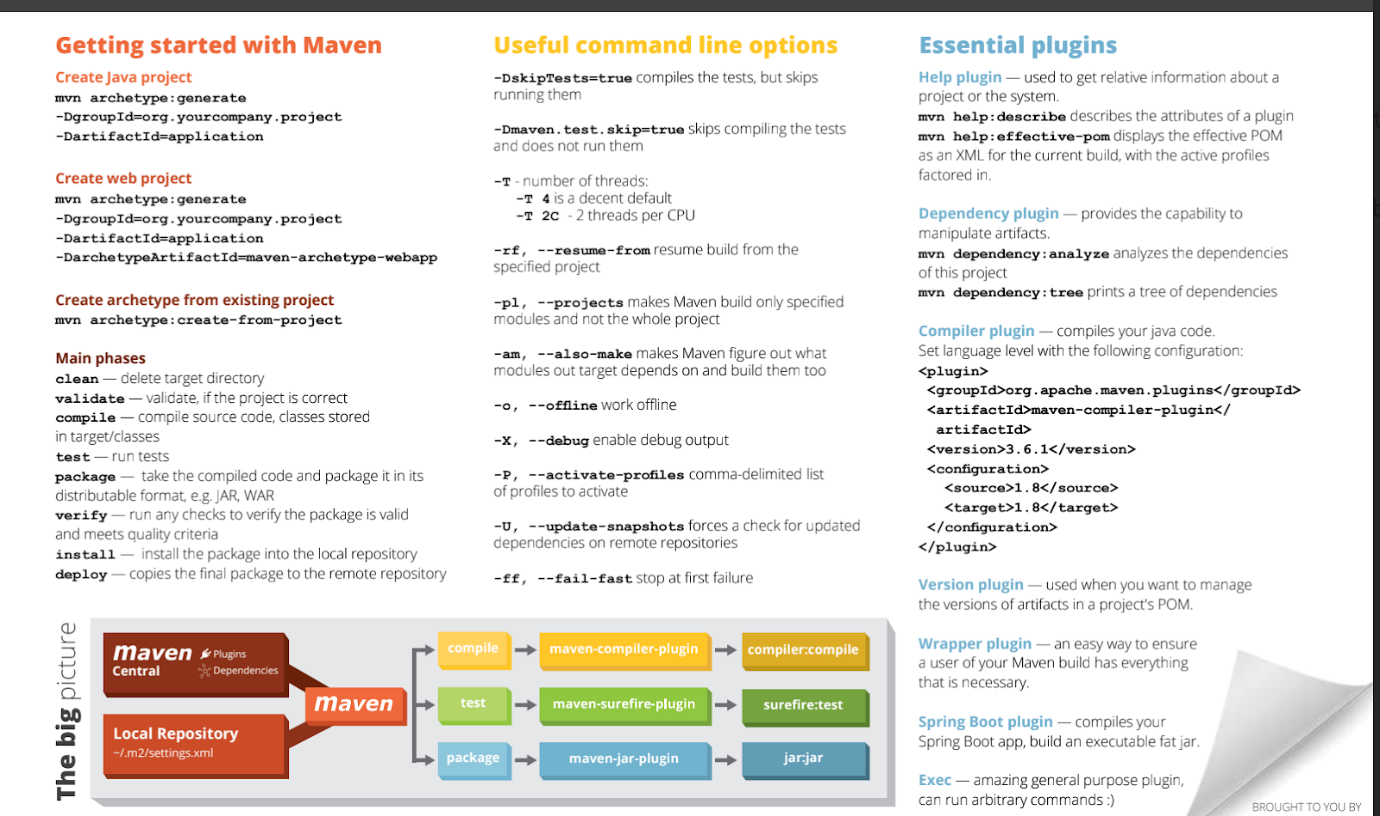
What is the Maven assembly plugin, and how is it used?

The Maven assembly plugin is used to create distributions or assemblies of your project, such as JARs with dependencies or ZIP files.

How do you deploy a Maven project to a remote repository?

To deploy a Maven project to a remote repository, use the mvn deploy command. Ensure that your settings.xml file has the necessary credentials and repository configuration.

CHEATSHEET



SCENARIO BASED REAL TIME

Question: How do you resolve dependency conflicts in a Maven project?

Answer: Maven uses the nearest definition strategy, where the version of the dependency declared closest to the project in the dependency tree takes precedence. Understanding the dependency hierarchy and utilizing the dependency:tree command helps identify and resolve conflicts.

Question: Explain how you would perform an offline build using Maven.

Answer: Maven allows offline builds by using the -o or --offline option. This prevents Maven from attempting to connect to remote repositories, relying on the locally cached dependencies. This is useful in scenarios with limited or no internet connectivity.

Question: Can you customize the Maven build lifecycle, and if so, provide an example?

Answer: Yes, the build lifecycle can be customized using plugins. For instance, the exec-maven-plugin allows executing arbitrary commands during the build process. Configuring it in the <build> section of the POM file enables the execution of custom scripts or commands.

Question: What are Maven snapshot releases, and when should they be used?

Answer: Snapshot releases are versions of a project that are in development. They have a unique version number with the "-SNAPSHOT" suffix. Snapshots are useful during development to allow continuous integration systems to pick up the latest changes. However, they should be avoided in release versions.

Question: How do you handle a multi-module Maven project, and what benefits does it offer?

Answer: In a multi-module project, Maven can build multiple projects within the same hierarchy. Each module represents a different subproject, and the parent POM coordinates the build process. This structure promotes code organization, reuse, and simplifies the management of complex projects.