# Maven Interview Questions and Answers

## 1. What is Maven, and why is it used in DevOps?

Maven is a build automation and dependency management tool primarily used for Java projects. It helps automate project compilation, packaging, testing, and deployment. In DevOps, Maven plays a key role in CI/CD pipelines, ensuring reliable builds and dependency management.

## 2. Explain the Maven Build Lifecycle.

Maven has three main build lifecycles:  
1. Clean: Cleans the project by removing compiled files (`mvn clean`).  
2. Default (Build): Handles compilation, testing, packaging, and deployment.  
3. Site: Generates project documentation (`mvn site`).  
  
Each lifecycle consists of phases such as validate, compile, test, package, install, and deploy.

## 3. What is a POM.xml file, and what are its key elements?

The Project Object Model (POM.xml) is the configuration file in Maven that contains:  
- Project Information: <groupId>, <artifactId>, <version>  
- Dependencies: Libraries required for the project  
- Plugins: Tools for compilation, testing, packaging  
- Build Configurations: Lifecycle management

## 4. How does Maven manage dependencies?

Maven uses the <dependencies> section in POM.xml to manage required libraries. Dependencies are downloaded from Maven repositories and stored in the local cache.

## 5. What is the difference between Maven and Ant?

Maven is declarative and uses POM.xml for configuration, whereas Ant is procedural and requires manual scripting.  
Maven has built-in dependency management, whereas Ant does not.

## 6. What are the different types of Maven repositories?

1. Local Repository: ~/.m2/repository stores dependencies downloaded from remote repositories.  
2. Central Repository: The default repository (Maven Central) for fetching dependencies.  
3. Remote Repository: Nexus, Artifactory used for team collaboration.

## 7. How do you add an external dependency in Maven?

Add the dependency in the <dependencies> section of POM.xml and run `mvn install`.

## 8. What is the scope of dependencies in Maven?

Scopes include compile, provided, runtime, test, and system.

## 9. What is the difference between SNAPSHOT and RELEASE versions?

SNAPSHOT versions are work-in-progress and change frequently, while RELEASE versions are stable for production use.

## 10. How can you exclude a transitive dependency in Maven?

Use the <exclusions> tag inside a dependency in POM.xml.

## 11. What is the role of the Maven Compiler Plugin?

It compiles Java source code and allows version control.

## 12. How do you clean and build a Maven project?

Use `mvn clean install` to remove compiled files and build artifacts.

## 13. What are Maven Goals and Phases?

Goals are specific tasks like compile, test, and package. Phases include multiple goals.

## 14. What is the command to skip test cases while building?

Use `mvn install -DskipTests`.

## 15. How can you create a JAR or WAR file using Maven?

Run `mvn package` to create a JAR or WAR file based on POM.xml configuration.

## 16. What is a Maven Profile, and how do you use it?

Profiles allow environment-specific configurations. Run `mvn install -Pdev` to activate a profile.

## 17. How do you deploy an artifact to Nexus or Artifactory?

Use `mvn deploy` after configuring distribution management in POM.xml.

## 18. How do you handle version conflicts in Maven?

Use dependency management or specify a particular version in POM.xml.

## 19. What is the difference between mvn install, mvn package, and mvn deploy?

- mvn package: Creates a JAR/WAR file.  
- mvn install: Stores artifacts locally.  
- mvn deploy: Uploads artifacts to a remote repository.

## 20. What is the difference between Parent POM and Dependency Management?

Parent POM is used for inheritance, whereas Dependency Management defines dependency versions without enforcing inheritance.