## ****1. List the differences between ANT and Maven.****

|  |  |
| --- | --- |
| **Ant** | **Maven** |
| Ant doesn’t have formal conventions, so we need to provide information about the project structure in the build.xml file. | Maven has a convention to place source code, compiled code, etc. So we don’t need to provide information about the project structure in pom.xml file. |
| Ant is procedural, you need to provide information about what to do and when to do through code. You need to provide order. | Maven is declarative, everything you define in the pom.xml file. |
| There is no life cycle in Ant. | There is a life cycle in Maven. |
| Ant is a toolbox. | Maven is a framework. |
| It is mainly a build tool. | It is mainly a project management tool. |
| The ant scripts are not reusable. | The maven plugins are reusable. |
| It is less preferred than Maven. | It is more preferred than Ant. |

**NOTE**: This is a common Maven Interview Question that you must know.

## ****2. What is Maven?****

[Maven](https://maven.apache.org/)is a project management and comprehension tool. Maven provides developers a complete build lifecycle framework. The development team is easily able to automate the project’s build infrastructure in almost no time as Maven uses a standard directory layout and a default build lifecycle.

## ****3. What are the main features of Maven?****

Some of the main features of Maven are:

* + **Simple to use**: Maven provides easy project settings that are based on genuine practices.
  + **Fast**: You can receive a fresh project or module that began in fewer seconds in Maven.
  + **Easy to learn:** Maven usage and commands are pretty easy to learn across all projects. Therefore ramp-up time for new developers coming onto a project is very less.
  + **Dependency management:** Maven provides superior dependency management including automatic updates and transitive dependencies.
  + **Multiple Projects**: You can easily work on multiple projects at the same time by using Maven.
  + **Huge Library:** Maven has a large and growing repository of libraries and metadata to use out of the box.

## ****4. 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

## ****5. Why should one use Maven?****

* It aids in setting up the project very quickly and it avoids complicated build files like build.xml. the pom.xml file is at the core of Maven.POM.xml is a collection of dependencies of your Java Project which one can specify to Maven. After this Maven will download all of them from the internet and store it to some repository i.e. local repository, central repository, and remote repository.
* It helps to bundle all the jars in your package i.e. in your War file or Ear file because all of them will be stored in the repository. So next time wherever you install this application that repository will be used for any dependencies lookup. In this way, your deployment file will be very light.

## ****6. Mention the steps for installing Maven on windows.****

Maven can be downloaded and installed on windows, linux, and MAC OS platforms. To install Maven on windows, you need to perform the following steps:

* + Download Maven and extract it.
  + Add JAVA\_HOME and MAVEN\_HOME in the list of environment variables.
  + Add the environment path in Maven variable.
  + The last step is the verification of Maven by checking its version.

## ****7. What does it mean when you say Maven uses Convention over Configuration?****

* + In the case of Configuration, developers have to create the build processes manually and they have to specify each and every configuration in detail. But, Maven uses convention where the developers need not create the build processes manually.
  + Also, for convention, users do not need to specify the configuration in detail. Once a developer creates a project in Maven then Maven will automatically create a structure. Developers just have to place the files appropriately. There is no need to specify any configuration details in pom.xml file.

**NOTE**: This is an important Maven Interview Question that you must know.

## ****8. What are the aspects Maven manages?****

Maven provides developers ways to manage following −

* + Builds
  + Documentation
  + Reporting
  + Dependencies
  + SCMs
  + Releases
  + Distribution
  + mailing list

## ****9. How do you know the version of mvn you are using?****

Type the following command – mvn -version

## ****10. What is POM?****

POM stands for Project Object Model. In Maven, it is a fundamental Unit of Work and it is an XML file. You can find it in the base directory of the project. It consists of information about the project and various configuration details used by Maven to build the project(s).

## ****11. What information does POM contain?****

POM contains the following configuration information −

* project dependencies
* plugins
* goals
* build profiles
* project version
* developers
* mailing list

## ****12. What is Maven artifact?****

* An artifact is a file, normally a JAR that gets deployed to a Maven repository. A Maven build creates one or more artifacts, such as a compiled JAR and a source JAR.
* Each artifact consists of a group ID, an artifact ID, and a version string. The three together uniquely identify the artifact. A project’s dependencies are specified as artifacts.

## ****13. What is the Maven Build lifecycle?****

A Build Lifecycle can be defined as a well-defined sequence of phases. It clearly defines the order in which the goals are to be executed. Each build phase contains a sequence of goals. If one life cycle is executed, all build phases in that life cycle are executed. If a build phase is executed, all build phases before it in the pre-defined sequence of build phases are executed.

**NOTE**: This is another important Maven Interview Question that you must know.

## ****14. Name the 3 build lifecycle of Maven.****

The three build lifecycles are −

**clean**: cleans up artifacts created by prior builds.

**default**: used to build the application.

**site**: generates site documentation for the project.

## ****15. What is the command to build your Maven site?****

Type the command − mvn site

## ****16. What would the command mvn clean do?****

This command deletes the target directory with all the build data before starting the build process.

## ****17. What are the different phases of a Maven Build Lifecycle?****

Following are the phases of Maven build lifecycle −

**validate** − validate the project and check if everything is correct and all necessary information is available.