Maven

->> What is Maven ?
Build automation tool for overall project management.
It helps in
1. checking a build status
2. generating reports (basically javadocs)
3. setting up the automated build process and monitors the same.

Why Maven ?
It eases out source code compilation, distribution, documentation, collaboration with different teams .
Maven tries 2 describe
1. How a software is built.
2. The dependencies, plug-ins & profiles that the project is associated in a standalone or a distributed environment.

Ant disadvantages
1.While using ant , project structure had to be defined in build.xml. Maven has a convention to place source code, compiled code etc. So no need to provide informationabout the project structure in pom.xml file.
2.Maven is declarative, everything you define in the pom.xml file. No such support in ant.
3.There is no life cycle in Ant, where as life cycle exists in Maven.

-> Maven advantages
4. Managing dependencies
5. Uses Convention over configuration - configuration is very minimal
6. Multiple/Repeated builds can be achieved.
7. Plugin management.
8. Testing - ability to run JUnit and other integration test suites.

> What is POM? (Project Object Model)
It is the core element of any maven project.
Any maven project consists of one configuration file called pom.xml.
Location --In the root directory of any maven project.
It contains the details of the build life cycle of a project.
Maven reads the pom.xml file, then executes the goal.

Contents :
Dependencies used in the projects (Jar files)
Plugins used

Project version
Developers involved in the project
Build profiles etc.

Elements of maven pom.xml file

1. project   It is the root element of pom.xml file.
2. modelVersion It is the sub element of project. It specifies the modelVersion.
3. groupId   It is the sub element of project. It specifies the id for the project group.(typically organization name)
4. artifactId   It is the sub element of project. It specifies the id for the artifact (project). An artifact is something that is either produced or used by a project. Examples of artifacts produced by Maven for a project include: JARs, WARs.
5. version   It is the sub element of project. It specifies the version of the artifact under given group.
6. packaging   defines packaging type such as jar, war etc.
7. name   defines name of the maven project.
8. plugins   compiler plugins , eclipse plugins
9. dependencies collection of dependencies for this project.
Within that -- dependency defines a specific dependency.(eg : hibernate dependency,spring web)
9. scope --   defines scope for this maven project. It can be compile, provided, runtime, test and system.

Goals in Maven
Goal in maven is nothing but a particular task which leads to the compiling, building and managing of a project. A goal in maven can be associated to zero or more build phases. Only thing that matters is the order of the goals defined for a given project in pom.xml. Because, the order of execution is completely dependent on the order of the goals defined.
eg : clean , build ,install ,test

What is a Maven Repository
A maven repository is a directory of packaged JAR file with pom.xml file. Maven searches for dependencies(JARs) in the repositories. There are 3 types of maven repository:
    Local Repository
    Central Repository
    Remote Repository

Maven searches for the dependencies in the following order:
Local repository then Central repository then Remote repository.

----

maven repositories
If dependency is not found in these repositories, maven stops processing and throws an error.

1. Maven Local Repository
Maven local repository is located in the file local system. It is created by the maven when you run any maven command.
By default, maven local repository is HOME / .m2 directory.
(Can be updated by changing the MAVEN_HOME/conf/settings.xml)

2. Maven Central Repository
Maven central repository is located on the web(Created by the apache maven community)
The path of central repository is: https://mvnrepository.com/repos/central

3) Maven Remote Repository
Maven remote repository is also located on the web. Some of libraries that are missing from the central repository eg JBoss library , Oracle driver etc, can be located from remote repository.

-> Maven Build Life Cycle What is it ?
The sequence of steps which is defined in order to execute the tasks and goals of any maven project is known as build life cycle in maven.

Maven comes with 3 built-in build life cycles
Clean - this phase involves cleaning of the project (for a fresh build & deployment)
Default - this phase handles the complete deployment of the project
Site - this phase handles the generating the java documentation of the project.

-> Build Profiles in Maven
It is a subset of elements which allows to customize builds for particular environment. Profiles are also portable for different build environments.
Build environment basically means a specific environment set for production and development instances. When developers work on development phase, they use test database from the production instance and for the production phase, the live database will be used.
So, in order to configure these instances maven provides the feature of build profiles. Any no. of build profiles can be configured and also can override any other settings in the pom.xml
eg : profiles can be set for dev, test and production phases.