# **Maven**

POM: Basic fundamental unit of Maven

Parent POM/POMS

1. Basically, these are parent projects without code.  
2. Used by companies to define the set of libraries/versions, plugins they want their teams using.  
3. Can have dependencies, build plugins, variables definitions, and even their own parent POM, forming a chain.  
4. A great example is Spring Boot. You can extract it to create production-grade web services crazily fast.

5. Maven has one default parent pom called super pom which is inherited by all pom by default

Repository:

1. Local Repository: Maven local repository is a folder location on your machine. It gets created when you run any maven command for the first time. (.M2 repository)
2. Maven central repository is repository provided by Maven community. It contains a large number of commonly used libraries. When Maven does not find any dependency in local repository, it starts searching in central repository using following URL − <https://repo1.maven.org/maven2/>
3. Remote Repository: Sometimes, Maven does not find a mentioned dependency in central repository as well. It then stops the build process and output error message to console. To prevent such situation, Maven provides concept of Remote Repository, which is developer's own custom repository containing required libraries or other project jars. (**Generally Remote Repositories are defined in setting.xml)**
4. **Sequence of Searching: Local -> Central -> Remote**

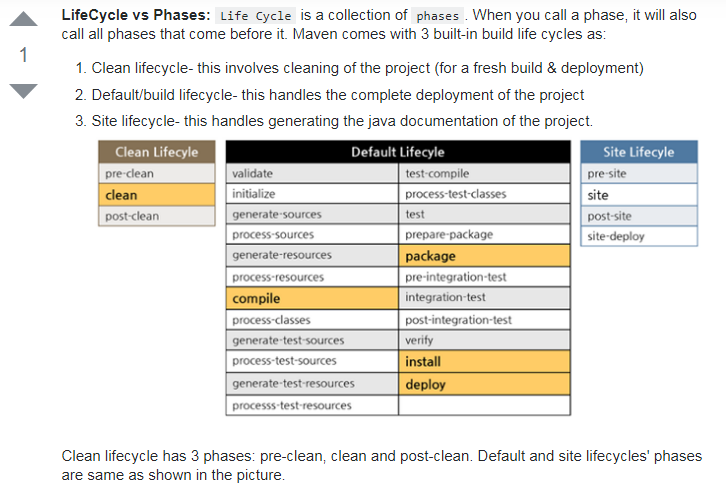
Difference between Central and Remote Repository

**Interview questions:**

<https://career.guru99.com/top-20-maven-interview-questions/>

**Maven Lifecycle**

<https://stackoverflow.com/questions/26607834/maven-lifecycle-vs-phase-vs-plugin-vs-goal/30953905>



**Plugin vs Dependency**

<https://stackoverflow.com/questions/11881663/what-is-the-difference-in-maven-between-dependency-and-plugin-tags-in-pom-xml>

Maven itself can be described as food processor which has many different units that can be used to accomplish different tasks. Those units are called plugins. For example, to compile your project maven uses maven-compiler-plugin, to run tests - maven-surefire-plugin and so on.

Dependency in terms of maven is a packaged piece of classes that your project depends on. It can be jar, war etc. For example, if you want to be able to write JUnit test, you'll have to use JUnit annotations and classes thus you have to declare that your project depends on JUnit.

**In my word:** Plugin is something which perform something, like compiler plugin compile the code, surefire plugin run the code. Dependency is something on which our code depends upon. Eg: To use selenium classes and functions we add selenium dependency to our project and it is placed in class path on compilation.