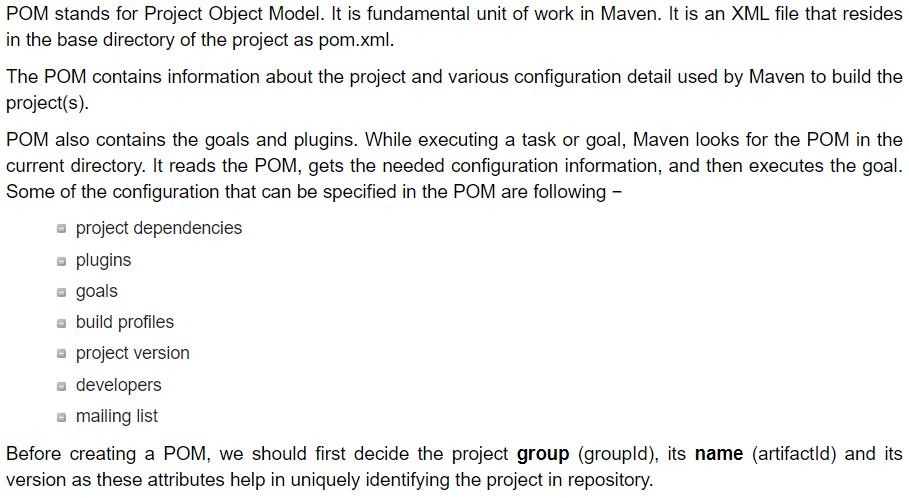
**POM.xml:**



**Maven coordinates:**

1. ArtifactId
2. groupId
3. version
4. package
5. When we execute a project in maven, it looks for the pom.xml in the project
6. <packaging> means the type of project. Like jar, war etc.
7. <name> for user friendly name
8. <URL> is optional. It tells which company owns the product
9. Groupid, artifactid, version, packaging is called maven coordinates

**groupId** uniquely identifies your project across all projects, so we need to enforce a naming schema. A groupId must follow Java's package name rules. This means it must start with a reversed domain name you control. You can create as many subgroups as you want.

**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.

**version** if you distribute it, then you can choose any typical version with numbers and dots (1.0, 1.1, 1.0.1, ...). Don't use dates as they are usually associated with SNAPSHOT (nightly) builds. If it's a third party artifact, you have to use their version number whatever it is, and as strange as it can look. For example,

**Super POM:**

