MAVEN

1. **What is Maven?**
   - A build tool and command prompt tool that called POM xml file that calls my runner class and manages my dependencies.
   - Maven is a build automation tool or a project management tool. With Maven we can import all libraries and can also create project structures. In Maven we have many inbuilt templates. These templates are called archetypes. A Maven is basically a tool used to compile our applications.
   - Command Prompt mvn archetype; generate
     - Creates project
   - Choose a # press enter
   - Choose a # press enter
   - GroupId; com.nameOfProject (usually a reversed domain name, like com.example.foo)
   - ArtifactID; testmavenproject
     - Version enter
     - Package enter
     - Y; enter

2. **Why Maven? How it helps you developing your project effectively?**
   - It helps to develop and managing project structure or applications like deployment, clean, packaging, jar and many more features for the Java-based project.
   - In another word, it is a Java tool. If you want to create a sample project or skeleton project you can use Maven. It is an automated build tool. The Maven focused on simplicity that it generates intelligent starters and assumes intelligence defaults. It also covers build-oriented phases in Application Lifecycle Management i.e. testing, deployment, builds management, and release versioning.
   - **It helps** to setup project very quickly and it avoids complicated build files like build.xml. Maven required files like POM.xml; it serves the purpose for Maven only. POM.xml is a collection of dependencies of your Java Project which one can specify to Maven and then Maven will download all of them from the internet and then store it to some repository i.e. local repository, central repository, and remote repository.

3. **What is Maven Artifact?**
   - An artifact is a file, usually a JAR, that gets deployed to a Maven repository.
   - A Maven build produces one or more artifacts, such as a compiled JAR and a "sources" JAR.
   - Each artifact has a group ID (usually a reversed domain name, like com.example.foo), an artifact ID (just a name), and a version string. The three together uniquely identify the artifact. Example:

```
<groupId>org.seleniumhq.selenium</groupId>
<artifactId>seleniumjava</artifactId>
<version>3.11.0</version>
```

   - A project's dependencies are specified as artifacts.

4. **Explain me the maven lifecycle?**
   - Commands can only run in the same directory where the specific **pom xml** file is located
   - 3 built in build lifecycles
     - Default → Handles your project deployment
     - Clean → Handles project cleaning
     - Site → Handles creation of project's site documentation