=> Maven :-

-> Maven is a powerful open-source "Project Automation Build Tool" i.e. it automates everything related to the building of project

-> Maven was developed by Jason van Zyl which was intially released in July 2004. Jason van Zyl founded the Apache Maven Project (Jakarta Turbine Project) which was later taken by Apache Software Foundation

-> Maven is written in java (so we must install JVM in order to work with maven)

-> Maven is mostly used with java projects but can also be used to build and manage the projecys written in other JVM based technologies i.e. Scala, Groovy, Kotlin etc

-> More build tools are :-

= Ant : Java, JVM based technologies, C/C++, JavaScript etc

= Gradle : Java, JVm based technologies, Android, Kotlin, C/C++ etc

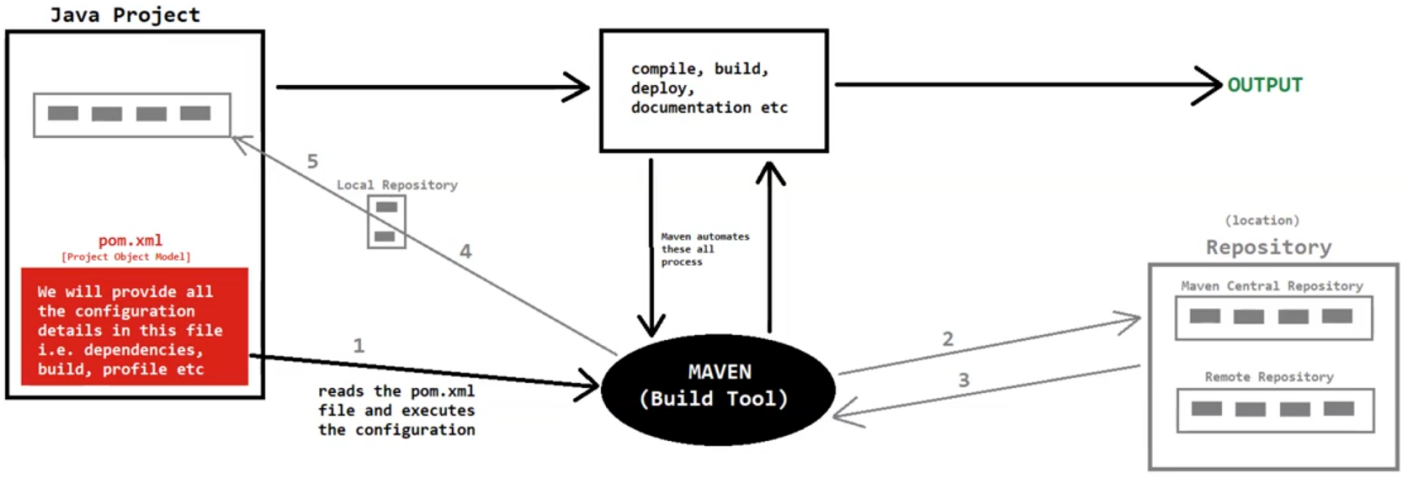
etc

-> Maven can perform following tasks :-

1. Creates the default project structure
2. Download the required dependencies (jar files)
3. Prepares the documentation
4. Compiles the source code
5. Packaging the project in jar, war or ear file
6. Install the packaged code on the server
7. Starts and stops the server
8. Build and deploy the project
9. Performs the test execution
10. Performs the test reports

etc

Working Of Maven



-> Advantages / Objectives of Maven :-
1. Makes the build process easy
2. Enhances the project performance
3. Easy to migrate to newer or older version
4. Strong error and integrity reporting
5. Integrate with "Version Control Systems" i.e. GIT etc etc

-> Working of Maven :-
= Diagram :-
= NOTE : Maven is controlled by "pom.xml" file

=> POM (Project Object Model) :-
    -> It is an xml file (pom.xml) which contains the information about the project and configuration details which is used (read) by Maven to build the project
    -> It is also known as "heart" of maven
    -> In maven1, this xml file was known as "project.xml" but from maven2 name was changed to "pom.xml"

    -> Syntax :-

-> Syntax :-

```
<?xml version="--" encoding="--" ?>

<project
    -----------
    ----------->

    <!-- configuration details -->
    1. Project Information
    2. SCM (Source Control Management)
    3. Properties
    4. Dependencies Configurations
    5. Build Settings Configurations
    6. Plugins and Goals Configurations
    7. Repositories
    8. Reporting Configurations
    9. Profiles Configurations


</project>
```

```
=> Project Information :-
    -> It contains the project details like version, groupid, artifactid, name, url etc
    -> Syntax :-
        <modelVersion>4.0.0</modelVersion>

        <groupId>com.example</groupId>
        <artifactId>my-project</artifactId>
        <version>1.0.0</version>

        <name>My Project</name>
        <description>A sample Maven project</description>
        <url>https://github.com/example/my-project</url>
```

=> Property References :-
->It provides the flexibility to the build tool to avoid hardcode values for eg. version number
-> Syntax :-

```
<properties>
    <java.version>1.8</java.version>
</properties>

<dependencies>
    <dependency>
        ----
        <version> ${java.version} </version>
    </dependency>
</dependencies>
```

=> Dependencies :-
-> Dependencies are the jar files or libraries which are required in our project
-> Maven will automatically download the dependencies which we have configured in "pom.xml" file. We dont need to download and add dependencies (jars) in our project manually
-> Syntax :

```
<dependencies>
    <dependency>
        <groupId>junit</groupId>
        <artifactId>junit</artifactId>
        <version>4.11</version>
        <scope>test</scope>
    </dependency>

    <dependency>
        <groupId>mysql</groupId>
        <artifactId>mysql-connector-java</artifactId>
        <version>8.0.28</version>
    </dependency>
</dependencies>
```

```
-----------------------------------

=> Repositories :-
    -> It is the location (server or local) from where maven will download and add the project required
    dependencies (jar file)
    -> There are mainly 3 main repositories :-
        1. Local Repository
            -> It is the location in our system (computer) where maven stores the dependencies
            -> Path : C:\Users\PC-Name\.m2\repository
        2. Maven Central Repository (Central Repository or Online Repository)
            -> It is default repository (public repository)
            -> It is the location on the server from where maven download the dependencies
            -> Path : https://repo.maven.apache.org/maven2/
            -> NOTE : While downloading the dependencies from Maven Central Repository, maven will also
            store these dependencies in local repository so that next time it can directly used.
        3. Remote Repository
            -> It is the location of any organization repository
            -> It is basicaly a private repository
            -> Path : https://organizationname.com/----
    -> Syntax :-
        <repositories>
            <repository>
                <id>central</id>
                <url>https://repo.maven.apache.org/maven2</url>
            </repository>
        </repositories>
```

=> What is difference between build, deploy and release ?

-> Build :-
= It is the process of converting the application compilation version to the executable (binary) version
= Build contains the following phases :-
- compilation
- linking
- production of distributed packages (installers)
- generating documentation
- execution of automated test
- generating reports
etc

-> Deploy :-
= Deploy is the process when we install the application on an environment for eg server

-> Release :-
= Release is the process of making the applicaiton available for end-users

=> Build Configurations :-
    -> It contains the configuration of the building process (build lifecycle) of an application
    -> There are 2 types of build configurations :-
        1. Plugin Configuration
        2. Resource Configuration

1. Plugin Configuration :-
   -> Maven is actually a collection of plugins which can be used to perform many tasks for eg :-
      = create jar or war or ear file
      = compile code files
      = unit testing of code
      = create project documentation
      = create project reports
        etc
   -> Maven has as internal framework i.e. "Maven Plugin Execution Framework" which is responsible to perform all above tasks
   -> There are 2 types of plugins :-
      1. Build Plugin
         = These plugins are executed during the build process i.e. clean, compile, deploy, install etc
         = These plugins are configured in <plugins> and <plugin> tag
      2. Reporting Plugin
         = These plugins are executed during the reporting phases i.e. issue tracking, project team, mailing lists etc
         = These plugins are configured in <reporting> tag

2. Resource Configuration :-

-> The resource plugin copy the files from input resource directory to output resource directory

-> For example, maven by default look for our project resources under src/main/resources directory. However many resources may not be available in src/main/resources directory, then we can specify those directories in resource configuration

-> Syntax :-

```
<resources>
    <resource>
        <directory> --- src/xyz --- </directory>
    </resource>
</resources>
```

```
=> What is archetype ?
    -> It is the project template or project model
    -> For simple maven project in java, we can select the archetype i.e. "maven-archetype-quickstart"
    and for web application we can select "maven-archetype-webapp" archetype


=> What is groupId ?
    -> The groupId is a unique identifier for a group or organization.
    -> It helps to distinguish your project from others, especially when projects from different sources
    might share the same artifactId
    -> It follows a reverse domain name pattern like ww.xyz.com -> com.xyz


=> What is artifactId ?
    -> It is the unique project name.
    -> Normally we can provide the name as :-
        = MavenFirstProject
        = mavenFirstProject
        = maven-first-project
```