```
============
Build Tools
============

=> Build tools are used to automate project build process.

=> Build process means convert project source code into executable format.

=> We have several build tools in the market

                        1) ANT (outdated)

                        2) Maven

                        3) Gradle

                        4) MS Build (Dot net)

                        5) NPM (angular, react, node js)

=============================
Java Project Execution Flow
=============================

=> Developers will develop source code for the project

                ex: .java files

=> We need to convert source code into byte code using java compiler.

=> When we compile source code it will be converted into byte code

                ex: .class files

=> We need to package .class files as jar or war file for execution.

                JAR : Java Archieve

                WAR : Web Archieve

=> Standalone apps will be packaged as jar file

=> Web Apps will be packaged as war file.

=======
Maven
=======

=> It is a build tool

=> It is free and open source s/w developed by Apache Org.

=> Maven s/w developed by using Java language.

=> Maven is used as java projects build automation tool.

Note: The main aim of maven is to automate and simplify java projects build process.

=================================
What we can do by using maven ?
=================================
1) We can create java project folder structure.

2) We can download required libraries

        ex: hibernate, spring, springboot, junit, logback.....

3) We can compile source code of the project

                .java file -------> .class file

4) We Execute Unit test cases of the project (junits)

5) We can package our application as jar or war file for deployment.


===========================
Maven Setup in Windows
===========================

@@@ Reference Video : https://www.youtube.com/watch?v=hV1OWzYpzxo

1) Download and Install Java

2) Setup JAVA_HOME and Java Path

3) Verify Java Installation using cmd

4) Download Maven Software from apache website

5) Setup MAVEN_HOME and Maven Path

6) Verify Maven Setup using cmd

===========================
Maven Setup in Linux
===========================

# check maven software availability
$ mvn -version

# install maven s/w
$ sudo yum install maven

====================
Maven Terminology
====================

1) archetype : Represents type of project we want to create

                        quick-start : stand-alone app (jar)

                        web-app : web application (war)

2) groupId : Represents company domain name

                        ex: in.ashokit

                            com.tcs

                            com.ibm

3) artifactId : Represents project name

                ex: sbi_car_app
                    ashokit_ecomm
```

```
4) version : Represents project version

                        SNAPSHOT means under development

                        RELEASE means delivered to client

5) packaging : Represents project executable format

                ex: jar or war

6) dependencies : Libraries requied for project development

                ex: hibernate, springboot, junit, log4j....

7) maven goals : To perform project build process

        ex: compile, test, package, install...

8) maven repositories : Location where maven dependencies will be stored.

        1) Central Repo (public, managed by apache)

        2) Remote Repo (private, company specific - nexus/jfrog)

        3) Local Repo (will be created in local system) (.m2)

=========================
Creating Maven Project
=========================

=> Execute below command in cmd to create maven stand-alone application

mvn archetype:generate -DgroupId=in.ashokit -DartifactId=my-app-1 -DarchetypeArtifactId=maven-
archetype-quickstart -DarchetypeVersion=1.5 -DinteractiveMode=false

=> Execute below command to create maven web application

mvn archetype:generate -DgroupId=in.ashokit -DartifactId=my-web-app -DarchetypeArtifactId=maven-
archetype-webapp -DarchetypeVersion=1.4 -DinteractiveMode=false

=> Navigate into project directory and execute maven goals

                $ mvn package -DskipTests=true


=============
Maven Goals
=============

=> Maven Goals are used to perform Project Build Process

        Syntax :  mvn <goal-name>

Note: We need to execute maven goals from project root directory
(where pom.xml is available)

=> We have several maven goals like below

# clean : To delete project target directory

# compile : Compile source code of the project

        Note: It will convert .java files to .class files
```

        Note: It will generate target directory to store .class files

# test : To execute project unit test code (junits)

                test = compile + test

# package : To package our project as a jar / war file

                package = compile + test + package

Note: application package will be stored into target directory.

                Ex: mvn clean package

===================
Maven Dependencies
===================

=> Maven dependencies means libraries required for the project development.

        Ex: spring-core, junit, hibernate etc..

Note: Developers will add required dependencies in project pom.xml file.

=> We can find maven dependencies in www.mvnrepository.com

```
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-core</artifactId>
    <version>6.1.13</version>
</dependency>
```

=> Add above dependency in project pom.xml file under <dependencies/> section and execute maven goals.

```
                $ mvn clean package
```

===================
Maven Repositories
===================

Repositoriy : It is a place where maven dependencies will be stored.

=> We have 3 types of repositories in maven

1) Local Repository

2) Central Repository

3) Remote Repository

=> Local Repository will be created in our machine (.m2 folder)

=> Central Repository will be maintained by apache organization (public)

=> Remote Repository will be maintained by our company to store shared libraries.

Note: To setup remote repositories we will use Nexus / JFrog softwares.

==================
Maven - Summary
==================

```
1) What is build tool & why

2) What is java application build process

3) Maven Introduction

4) Maven Setup in windows and linux

5) What we can do using maven

6) Maven Terminology

7) Maven Project creation

8) Maven Dependencies

9) Maven Goals

10) Maven Repositories
```