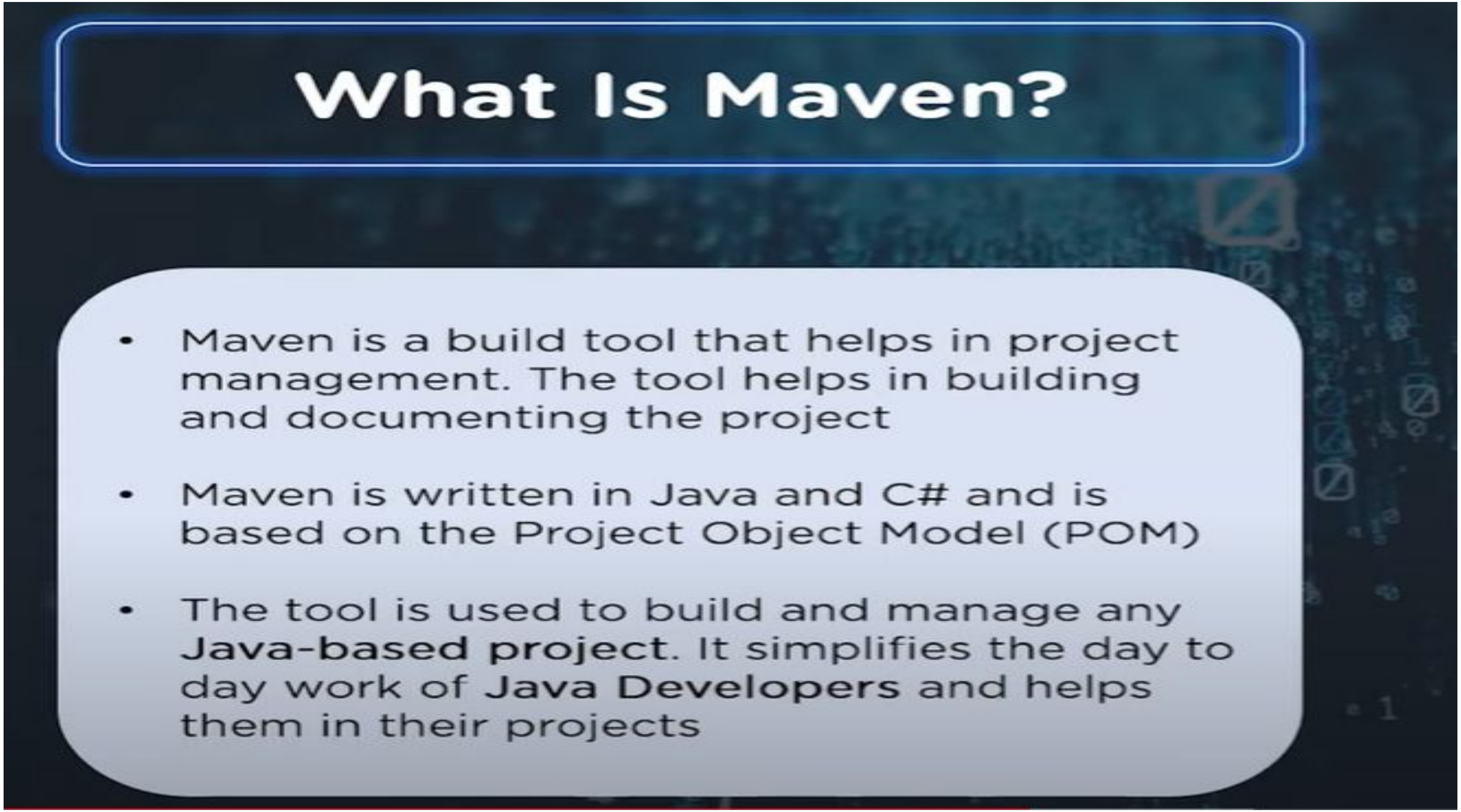


What Is Maven?

- Maven is a build tool that helps in project management. The tool helps in building and documenting the project
- Maven is written in Java and C# and is based on the Project Object Model (POM)
- The tool is used to build and manage any **Java-based project**. It simplifies the day to day work of **Java Developers** and helps them in their projects

The Problems That Maven Solved:

- Getting right JAR files for each project as there may be different versions of separate packages
- To download dependencies visiting of the official website of different software is not needed. We can just visit "mvnrepositiry.com"
- Helps to create the **right project structure** which is essential for execution

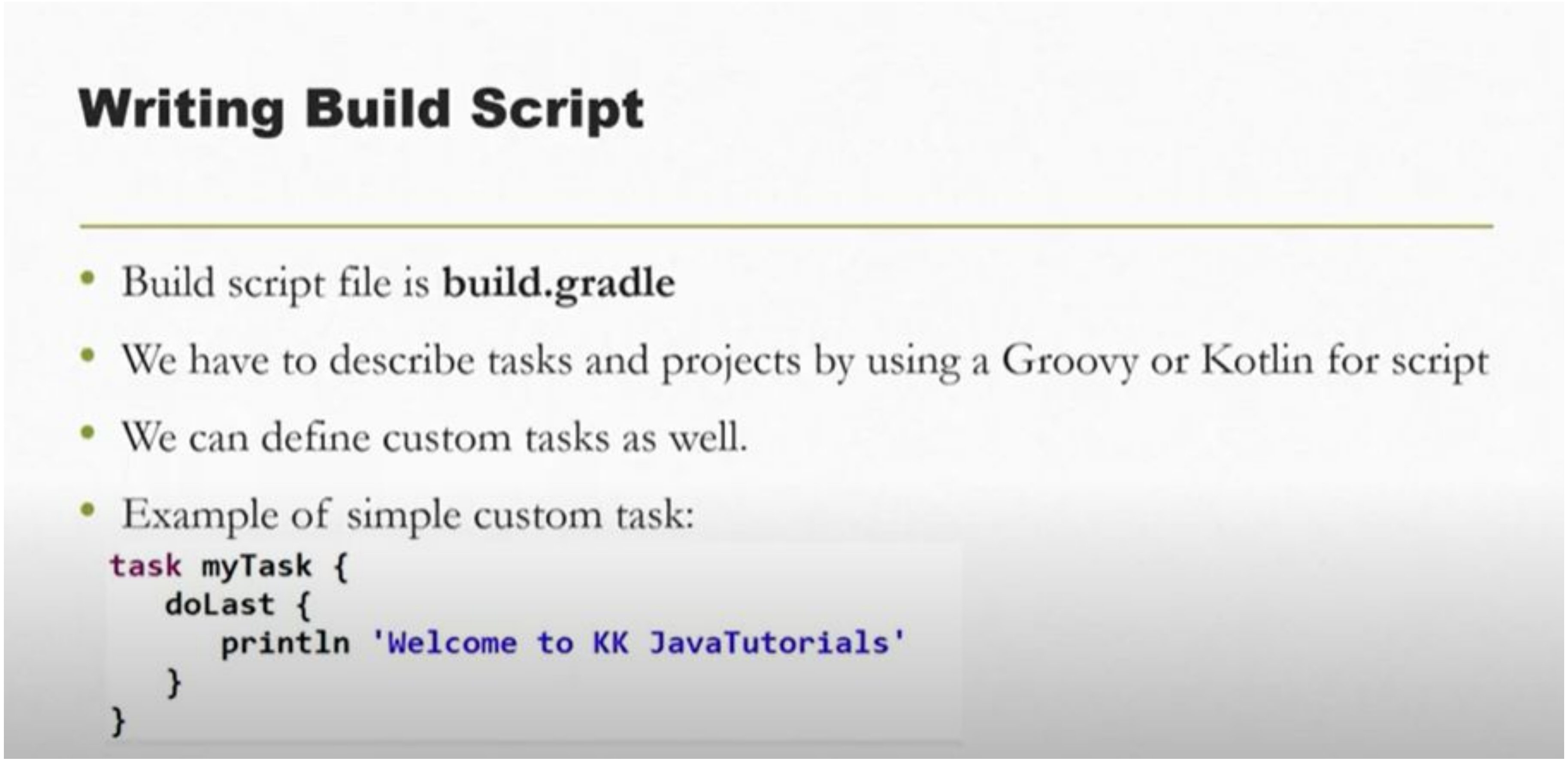
What is Gradle ?

- **Gradle** is an open source **build automation** tool that is based on the concept of **Apache Maven** and **Ant**.
- It introduces a **Kotlin and Groovy-based DSL(Domain Specific Language)** instead of XML (Extensible Markup Language) for declaring the project configuration.
- Gradle offers an elastic model that can help the development lifecycle from compiling and packaging code for web and mobile applications.
- It provides support for the **building**, **testing**, and **deploying software** on different platforms.
- It has been developed for building automation on many languages and platforms, including Java, Scala, Android, C / C ++, and Groovy. Gradle provides integration with several development tools and servers, including Eclipse, IntelliJ, Jenkins, and Android Studio.
- Some Leading Enterprise companies like **LinkedIn, Google** and **Netflix** use Gradle
- It is developed to **overcome the drawbacks of Maven and Ant** and supports a wide range of IDEs.

Why to Use Gradle ?

The following is the list of features that Gradle provides:

- Free and open-source
- High Performance
- Highly Customizable
- Performance
- Flexibility
- Multi-project build support
- Extensibility
- Incremental Builds
- Familiar with the Java
- Gradle Wrapper
- User Experience



Writing Build Script

- Build script file is **build.gradle**
- We have to describe tasks and projects by using a Groovy or Kotlin for script
- We can define custom tasks as well.
- Example of simple custom task:

```
task myTask {
   doLast {
      println 'Welcome to KK JavaTutorials'
   }
}
```

Gradle Projects and Tasks

- Everything in Gradle is based on the project and task
  1. Task
  2. Project

Gradle Task

Gradle

- ❑ Task is a single unit of work that a build performs.
- ❑ These tasks can be compiling classes, creating a JAR, Generating Javadoc, and publishing some archives to a repository and more.
- ❑ single atomic piece of work for a build process.

Types of Task

- There are two types of tasks in Gradle:
  - Default task
  - Custom task

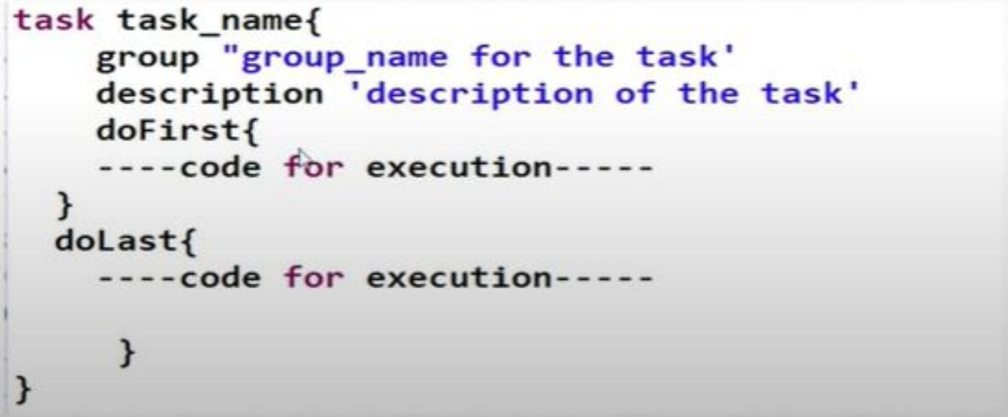
Default Tasks

- Default tasks are predefined tasks of Gradle.
- We can list the default tasks by running the gradle task commands.
  - $gradle tasks
  - $gradle tasks --all

Custom Tasks

Gradle

- Gradle allows us to create tasks and these tasks are called custom tasks. Custom tasks are user-defined tasks that are built to perform some specific work.



```
task task_name{
    group "group_name for the task'
    description 'description of the task'
    doFirst{
    ----code for execution-----
  }
  doLast{
    ----code for execution-----

      }
}
```

Gradle Repository

Gradle

- ❑ Specify the location of modules/libraries so that the gradle build can consume them. The location for storing modules/libraries is called a **repository**.
- ❑ Repositories can be in different forms, such as a local or remote repository.
- ❑ At runtime, Gradle will discover the declared dependencies required for operating a specific task. Once a dependency is resolved, the resolution mechanism stores the essential files of dependency in the local cache memory also call dependency cache.
- ❑ Future builds reuse the files saved in the cache to skip unnecessary network calls.

Gradle

Gradle supports the following repositories:

1. Ivy repositories
2. Maven repository
3. Flat directory repository

configure Maven central as dependency source.

- In your build file you specify the remote repositories to look for dependencies. Gradle supports Maven and Ivy repositories to search for dependencies. The following listing shows how to configure Maven central as dependency source.

```
repositories {
    mavenCentral()
}

 OR

repositories {
    maven {
        url "https://repo.maven.apache.org/maven2/"
    }
}
```

You can also specify other targets, for example Bintray as Maven repository

```
repositories {
    maven ("http://jcenter.bintray.com/")
}
```

Gradle Dependencies

- Gradle build script describes a process of building projects. Most of the projects are not self-contained. They need some files to compile and test the source files.
- For example, to use Spring, we must include some Spring JARs in the classpath. Gradle uses some unique script to manage the dependencies, which needs to be downloaded.
- Every dependency is applied to a specified scope. For example, some dependencies are used to compile the source code and some will be available at runtime.
- Gradle considers the outcomes of building and publishing the projects

A dependency can be used on different phases of the project. These phases can be:

- **Compile:** The dependencies required to compile the source of the project.
- **Runtime:** These dependencies are used at runtime by classes. By default, it also contains the compile-time dependencies.
- **Test Compile:** These dependencies are required to compile the test source of the project. It also contains the compiled classes and the compile-time dependencies.
- **Test Runtime:** These dependencies are required to run the tests. It also contains runtime and test compile dependencies.

Gradle vs. Maven

Gradle

Maven	Gradle
It is a software project management system that is primarily used for java projects	It is a build automation system that uses a Groovy or Kotlin based DSL (domain-specific language)
It uses an XML file for declaring the project, its dependencies, the build order, and its required plugin.	It does not use an XML file for declaring the project configuration.
It is based on the phases of the fixed and linear model.	It is based on a graph of task dependencies that do the work.
It does not use the build cache; thus, its build time is slower than Gradle.	It avoids the work by tracking input and output tasks and only runs the tasks that have been changed. Therefore it gives a faster performance.
Maven has a limited number of parameters and requirements, so customization is a bit complicated.	Gradle is highly customizable; it provides a wide range of IDE support custom builds.