Explain the working and the differences between Maven, Gradle and SBT in detail.

Maven:

* It is a “build in management tool”.
* And its main purpose is to see that how far the .java files were compiled as .class.

Which was packed as .jar or .war or .ear files

* Where the pre-processing or post-processing of tools take place
* CLASSPATH will be managed
* Other tasks takes place where process will be built.
* Apache Ant or Gradle or Makefiles which were used in C/C++ will do the similar process.
* So there is no need of any sort of additional tools since everything will be already available
* There will also be no need of installation or download of any libraries.
* It has "build portability" so there will be any issues when same code is run on one or more computer.

Gradle:

* It is another sort of method where all the features from build systems will be taken and combined as one.
* And improvement will be done based on the shortcomings in the future.
* It is a JVM system where one is capable of writing their own java program by making use of android studio.
* The best of all in gradle is that it is a plugin based system.
* Where one can write their own programming language and packages can be built certain packages like jar from certain sources where you can write a complete set of plugin in Java or Groovy, and can distribute it to rest of world.

SBT:

SBT is an open source build tool for Scala and Java projects, similar to Java's Maven or Ant.

Its main features are:

* native support for compiling Scala code and integrating with many Scala test frameworks
* build descriptions written in Scala using a DSL
* dependency management using Ivy (which supports Maven-format repositories)
* continuous compilation, testing, and deployment
* integration with the Scala interpreter for rapid iteration and debugging
* support for mixed Java/Scala projects
* It is Scala built in tool.

DIFFERENCES –

* SBT will be always focused mainly on scala for the purpose of dependency management.
* With the use of XML files one is capable to control the entire the software cycle.
* With the help of project object model a person is capable of intercepting the all the software setting will start from compile to test, packaging and deploy.
* In case of maven since it has XML syntax it suffers with bad issues and it has its own dependency manager.
* When one needs to write OM in maven it will be much expensive and will be annoying.
* In case of maven,Ant and Ivy gradle is the top most on .
* As well it makes use of mavel repositories.
* XML will not be used by mavel and it acts as a polyglot built tool which will combine Ant API with the Groovy language in order to help the developers to build the script with the help of intuitive DSL.
* With the help of groovy language one can do their own task when paired with mavel.
* Mavel will do the task like groovy but few lines were enough to code in gradle.
* But the plugin for mavel and groovy will be same which was used in your projects.