1. **JMS Provider** is the **messaging server** that **implements the JMS Specification**.
2. So, JMS Provider such as ActiveMQ, WebSphere MQ etc are MOM that we will be using when working with **Message-Oriented Application**.
3. **Apache ActiveMQ Artemis**:
   1. JMS Provider = Messaging Server
   2. To create client.
   3. Can be run on our local machine as well as in production.
4. Install Apache ActiveMQ Artemis and then create a Broker (JSM Provider).   
   Then our app can use all the messaging features through JMS API.  
   We will master JMS APIs through the course.
5. Apache ActiveMQ Artemis gives
   1. Durability.
   2. Scalability.
   3. Acknowledgement.
   4. Transaction Management.
   5. Clustering.
   6. and more.
6. Once installed, we can create administered objects (Called so as usually created by an administrator) which are put in JNDI registory.
   1. **ConnectionFactory.**
   2. **Queue**: For P2P Messaging.
   3. **Topic**: For Pub-Sub Messaging.
7. Once we created those above administrated objects, we can access them in our JMS Client Applications through JNDI (Java Naming and Directory Interface).
8. 