Congratulations! You have completed this module. At this point in the course, you know:

* The Data Science Task Categories include:
  + Data Management - storage, management and retrieval of data
  + Data Integration and Transformation - streamline data pipelines and automate data processing tasks
  + Data Visualization - provide graphical representation of data and assist with communicating insights
  + Modelling - enable Building, Deployment, Monitoring and Assessment of Data and Machine Learning models
* Data Science Tasks support the following:
  + Code Asset Management - store & manage code, track changes and allow collaborative development
  + Data Asset Management - organize and manage data, provide access control, and backup assets
  + Development Environments - develop, test and deploy code
  + Execution Environments - provide computational resources and run the code

The data science ecosystem consists of many open source and commercial options, and include both traditional desktop applications and server-based tools, as well as cloud-based services that can be accessed using web-browsers and mobile interfaces.

**Data Management Tools**: include Relational Databases, NoSQL Databases, and Big Data platforms:

* MySQL, and PostgreSQL are examples of Open Source Relational Database Management Systems (RDBMS), and IBM Db2 and SQL Server are examples of commercial RDBMSes and are also available as Cloud services.
* MongoDB and Apache Cassandra are examples of NoSQL databases.
* Apache Hadoop and Apache Spark are used for Big Data analytics.

**Data Integration and Transformation Tools:** include Apache Airflow and Apache Kafka.

**Data Visualization Tools:** include commercial offerings such as Cognos Analytics, Tableau and PowerBI and can be used for building dynamic and interactive dashboards.

**Code Asset Management Tools:** Git is an essential code asset management tool. GitHub is a popular web-based platform for storing and managing source code. Its features make it an ideal tool for collaborative software development, including version control, issue tracking, and project management.

**Development Environments:** Popular development environments for Data Science include Jupyter Notebooks and RStudio.

* Jupyter Notebooks provides an interactive environment for creating and sharing code, descriptive text, data visualizations, and other computational artifacts in a web-browser based interface.
* RStudio is an integrated development environment (IDE) designed specifically for working with the R programming language, which is a popular tool for statistical computing and data analysis.

Congratulations! You have completed this module. At this point in the course, you know:

* You should select a language to learn depending on your needs, the problems you are trying to solve, and whom you are solving them for.
* The popular languages are Python, R, SQL, Scala, Java, C++, and Julia.
* For data science, you can use Python's scientific computing libraries like Pandas, NumPy, SciPy, and Matplotlib.
* Python can also be used for Natural Language Processing (NLP) using the Natural Language Toolkit (NLTK).
* Python is open source, and R is free software.
* R language’s array-oriented syntax makes it easier to translate from math to code for learners with no or minimal programming background.
* SQL is different from other software development languages because it is a non-procedural language.
* SQL was designed for managing data in relational databases.
* If you learn SQL and use it with one database, you can apply your SQL knowledge with many other databases easily.
* Data science tools built with Java include Weka, Java-ML, Apache MLlib, and Deeplearning4.
* For data science, popular program built with Scala is Apache Spark which includes Shark, MLlib, GraphX, and Spark Streaming.
* Programs built for Data Science with JavaScript include TensorFlow.js and R-js.
* One great application of Julia for Data Science is JuliaDB.

Congratulations! You have completed this module. At this point in the course, you know that:

* Python offers a diverse library ecosystem for data science, covering scientific computing (Pandas, NumPy), visualization (Matplotlib, Seaborn), and high-level machine learning (Scikit-learn). These libraries offer tools for data manipulation, mathematical operations, and simplified machine learning model development.
* Application Programming Interfaces (APIs) facilitate communication between software components. REST APIs, specifically, facilitate internet communication and access resources like storage. Key API terms include client (user or code accessing it), resource (service or data), and endpoint (API's URL).
* Machine learning models analyze data and identify patterns to make predictions and automate complex tasks—the three fundamental types of machine learning are supervised, unsupervised, and reinforcement learning. Supervised learning includes regression and classification models for predictive modeling and pattern recognition. Deep learning, an advanced subset of machine learning, mimics the brain's processing, enabling intricate problem-solving in various domains.
* The Community Data License Agreement (CDLA) facilitates open data sharing by providing clear licensing terms for distribution and use, and the IBM Data Asset eXchange (DAX) site contains high-quality open data sets.
* The Model Asset eXchange (MAX) provides a wealth of pre-trained deep learning models, empowering developers with readily deployable solutions for various business challenges.