



# Introduction to Big Data

Data Boot Camp

Lesson 22.1



The background is a dark charcoal gray with a series of parallel diagonal lines running from the top-left to the bottom-right. Overlaid on this are several teal-colored geometric shapes: a large central triangle pointing right, a smaller triangle to its left, and a square to its right. Scattered around these shapes are various white line-art symbols, including a plus sign, a minus sign, a circle with a dot, a circle with a horizontal line, a circle with a vertical line, a circle with a diagonal line, a circle with a zigzag line, a circle with a dot, a circle with a horizontal line, a circle with a vertical line, a circle with a diagonal line, a circle with a zigzag line, a circle with a dot, a circle with a horizontal line, a circle with a vertical line, a circle with a diagonal line, and a circle with a zigzag line.

**WELCOME**

# Class Objectives

---

By the end of this lesson, you will be able to:



Identify the parts of the Hadoop ecosystem.



Write a Python script that implements the MapReduce programming model.



Identify the differences between the Hadoop and Spark environments.



Create a DataFrame by using PySpark.



Filter and order a DataFrame by using Spark.

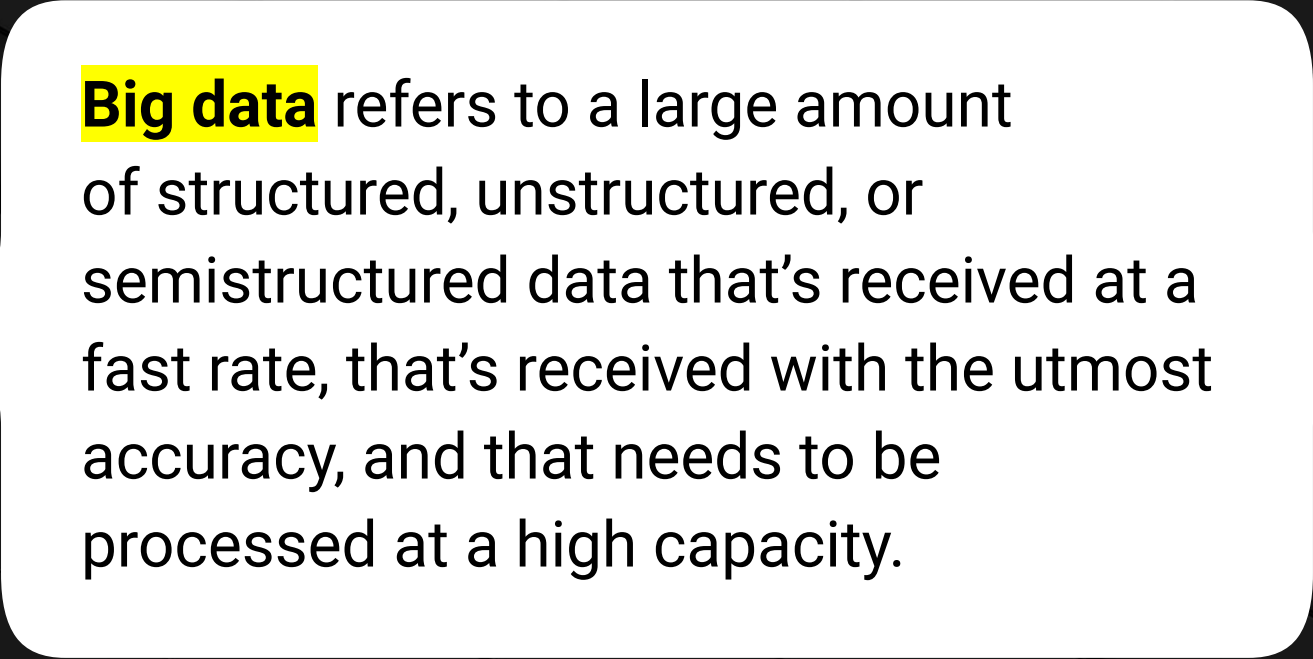
# Intro to Big Data



# Instructor Demonstration

---

## Intro to Big Data



**Big data** refers to a large amount of structured, unstructured, or semistructured data that's received at a fast rate, that's received with the utmost accuracy, and that needs to be processed at a high capacity.

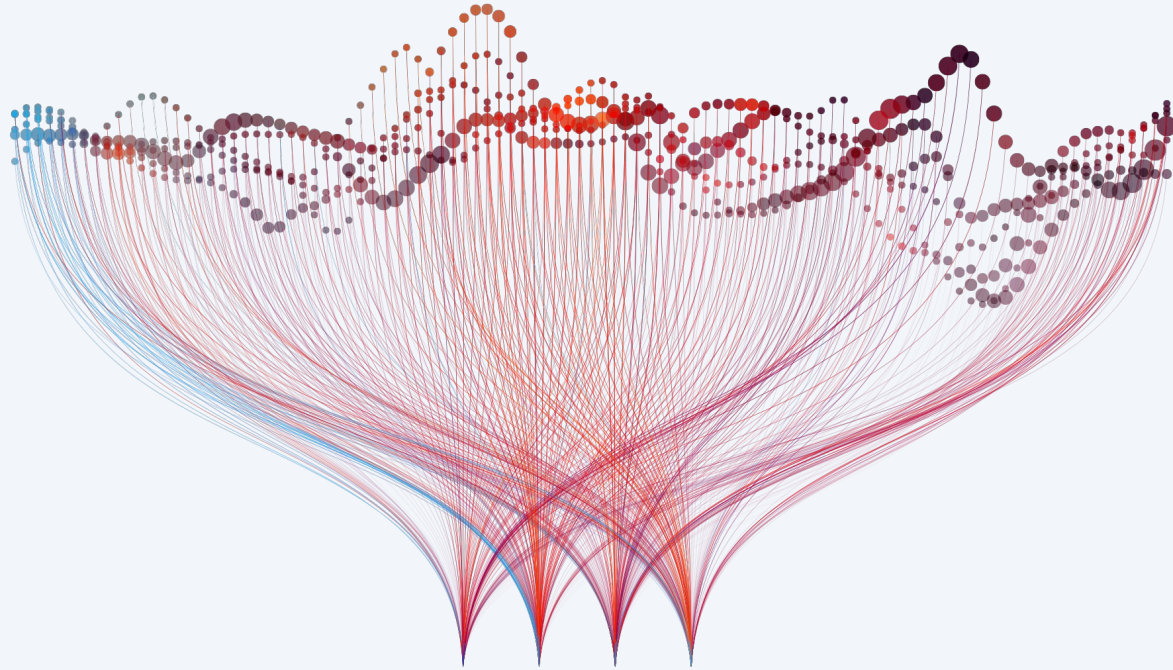


**Large datasets have been on the rise since the inception of the internet. But, what's a large dataset?  
And, just how big is big data?**

# Big Data: History

---

In the late 1990s, the term “big data” referred to any information that filled more than a gigabyte. Today, big data can range from hundreds of gigabytes to exabytes of data.





# How Big Is Big Data?

---

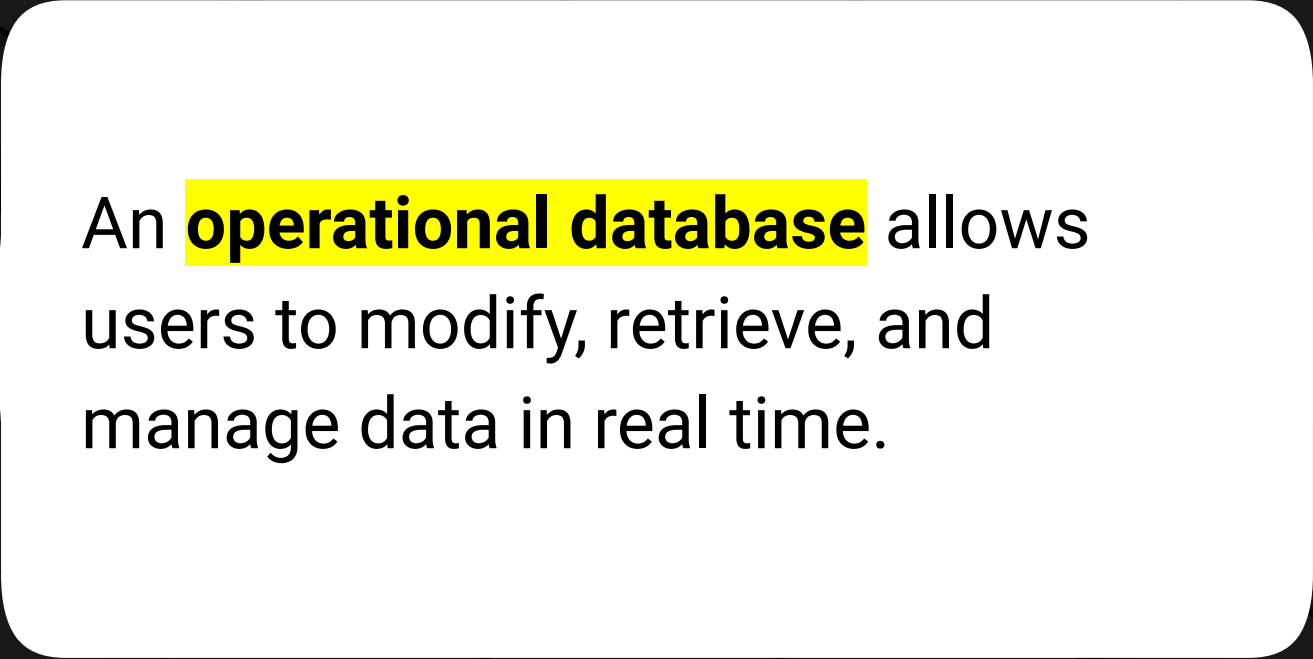
To give us an idea of just how big these amounts are, the following table lists data sizes along with an example of what each size might contain:

Size	Example
One kilobyte (KB)	A paragraph in a text document
One megabyte (MB)	A small novel
One gigabyte (GB)	Either about 10,000 rows in a CSV file or one low-definition streamed movie
One terabyte (TB)	About 500 hours worth of videos on the internet
One petabyte (PB)	About 11,000 high-definition movies
One exabyte (EB)	About 11 million high definition movies

**NETFLIX** has about 100 petabytes worth of stored videos across the world.



**Generally, we consider a dataset to be big data if it's too large for an operational database.**



An **operational database** allows users to modify, retrieve, and manage data in real time.

# The Four Vs of Big Data

---

01

**Volume:** The size of the data.

02

**Velocity:** The rate at which the data is received.

03

**Variety:** The diversity of the data: structured, unstructured, or semistructured.

04

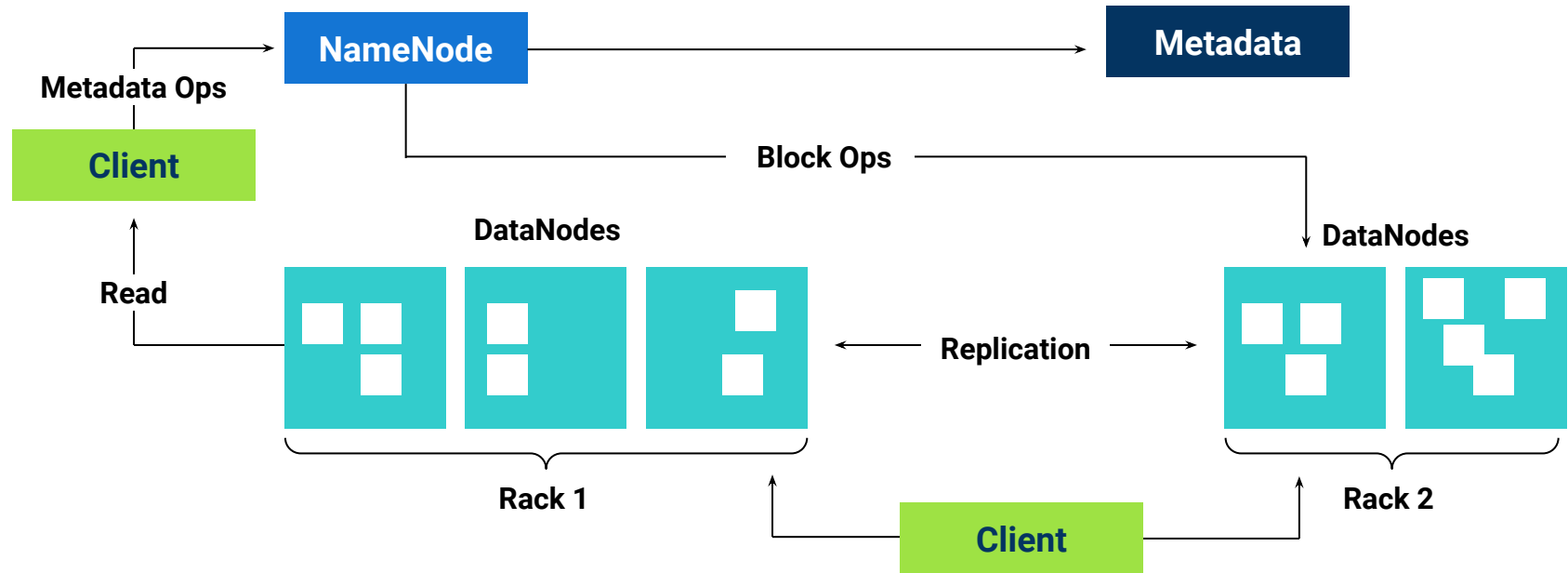
**Veracity:** The uncertainty of the data. That is, how accurate is the data? Does it include errors, abnormalities, or biases?

**Hadoop** is an open-source  
framework for big data.  
It consists of several components.



# Hadoop Distributed File System

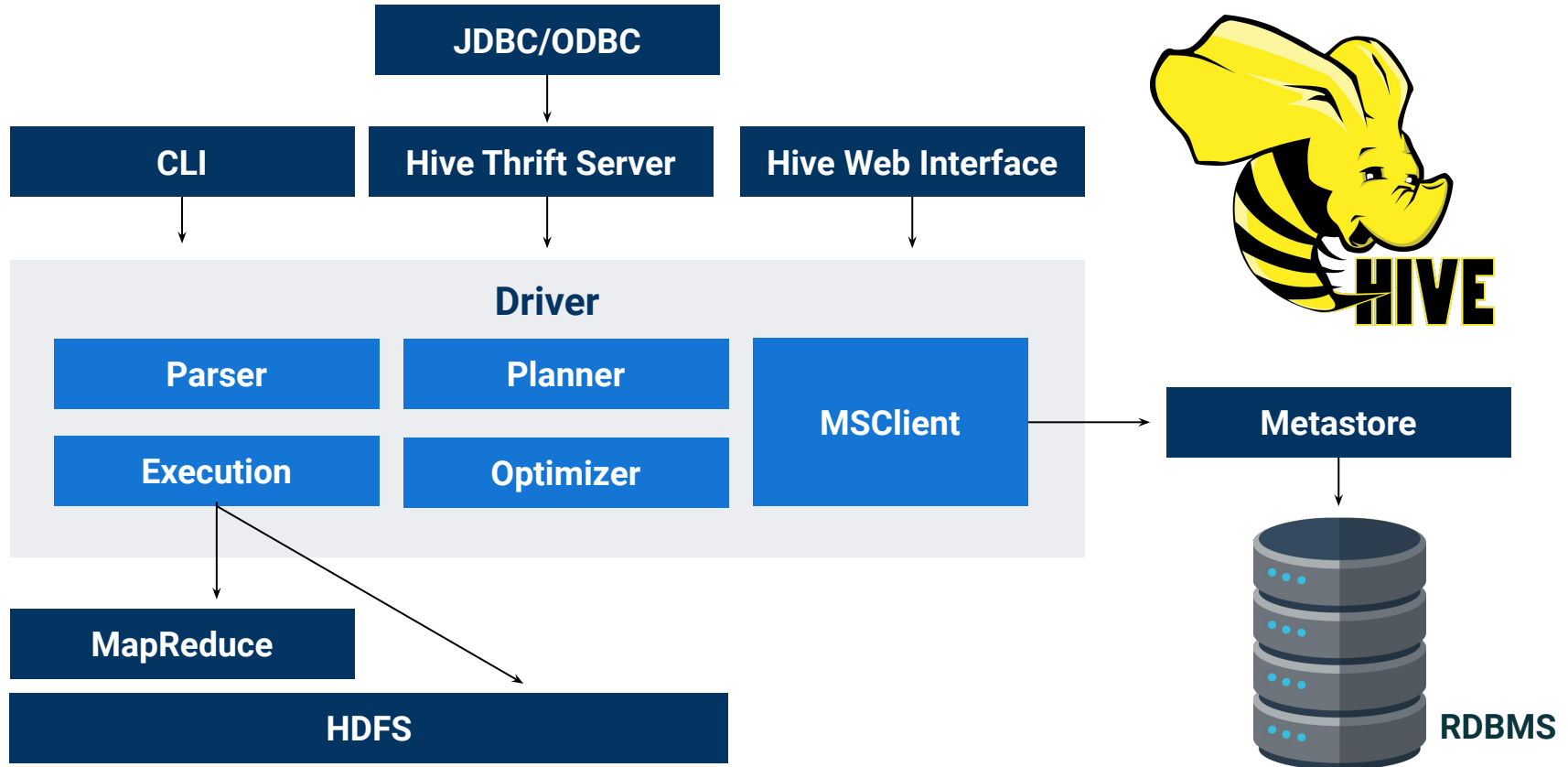
The **Hadoop Distributed File System (HDFS)** enables the efficient and cheap storage of large quantities of data across multiple servers while minimizing the risk of data loss.



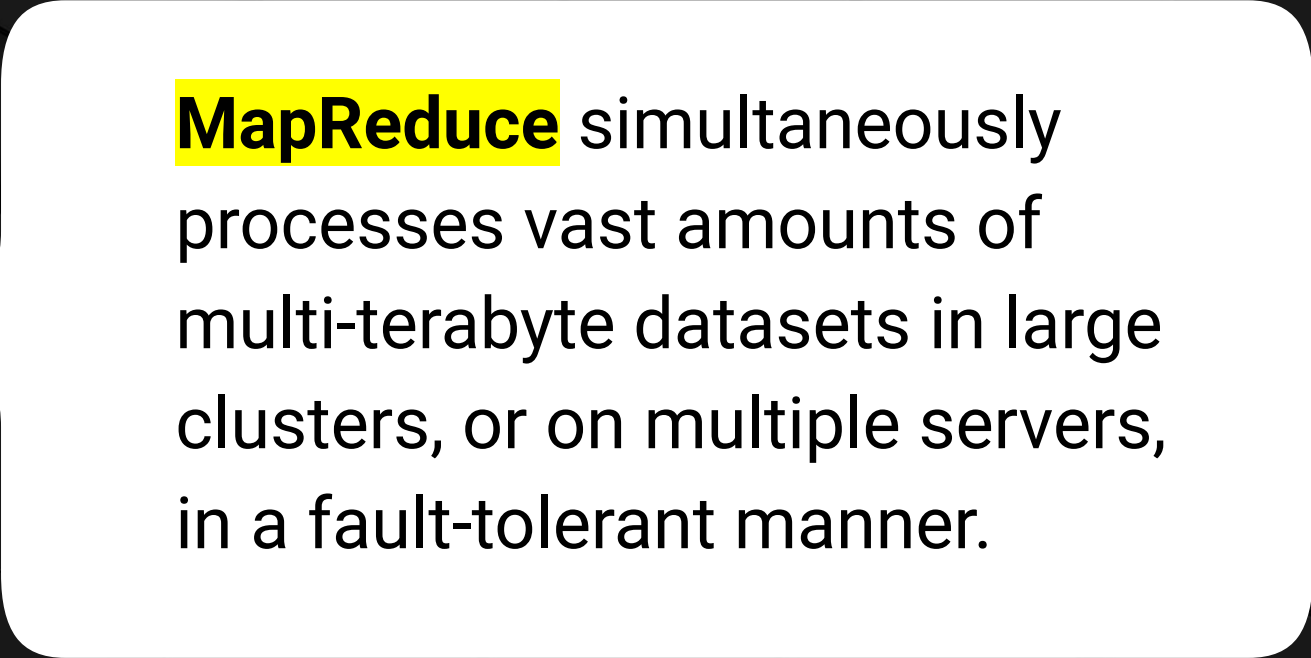
**Hive** is a SQL-like query tool for big data.



# Hive Architecture







**MapReduce** simultaneously processes vast amounts of multi-terabyte datasets in large clusters, or on multiple servers, in a fault-tolerant manner.

# Questions?



# MapReduce with MRJob



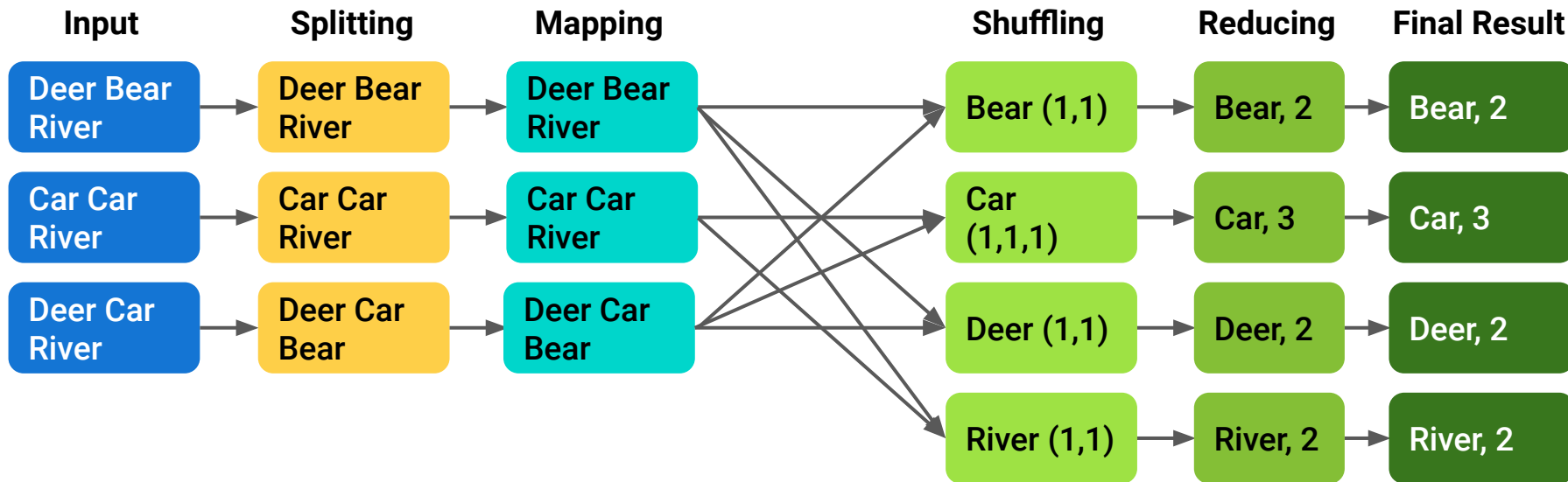
# Instructor Demonstration

---

## MapReduce with MRJob

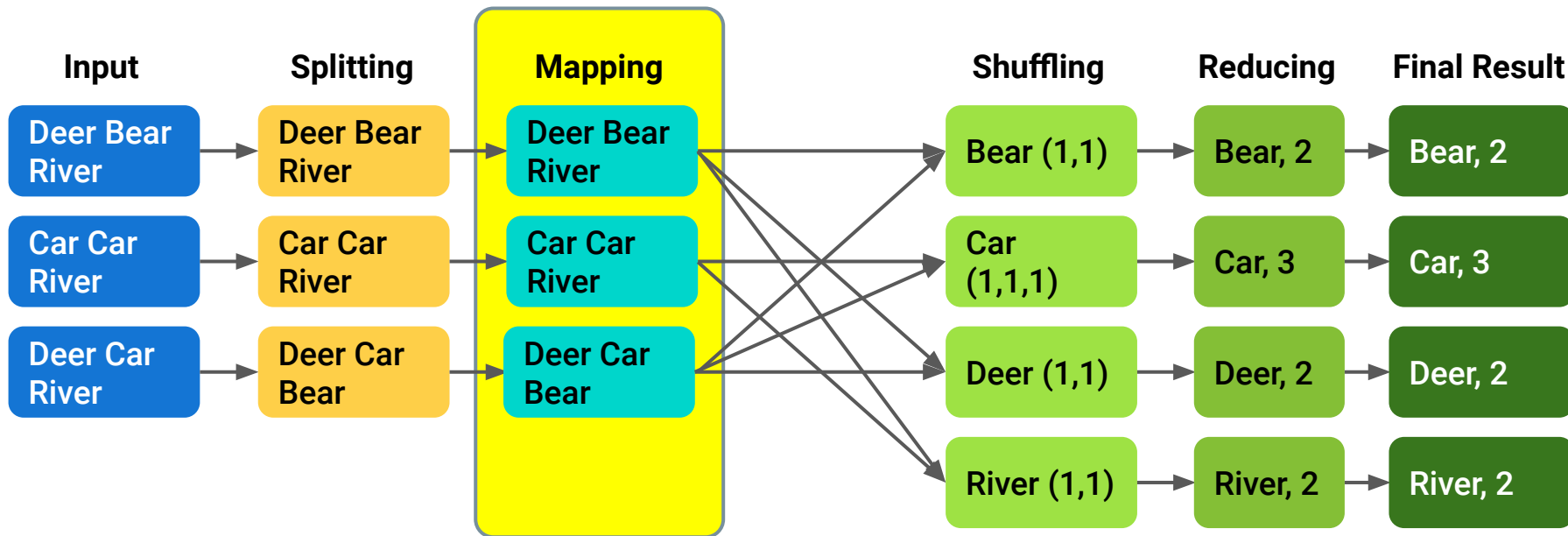
# MapReduce with MRJob (1 of 3)

Google created MapReduce with the initial purpose of indexing all the information on the internet. Now, people use MapReduce as a way to distribute and process the data in a cluster. Basically, a MapReduce job divides the input data into chunks, which the map tasks process in a parallel manner.



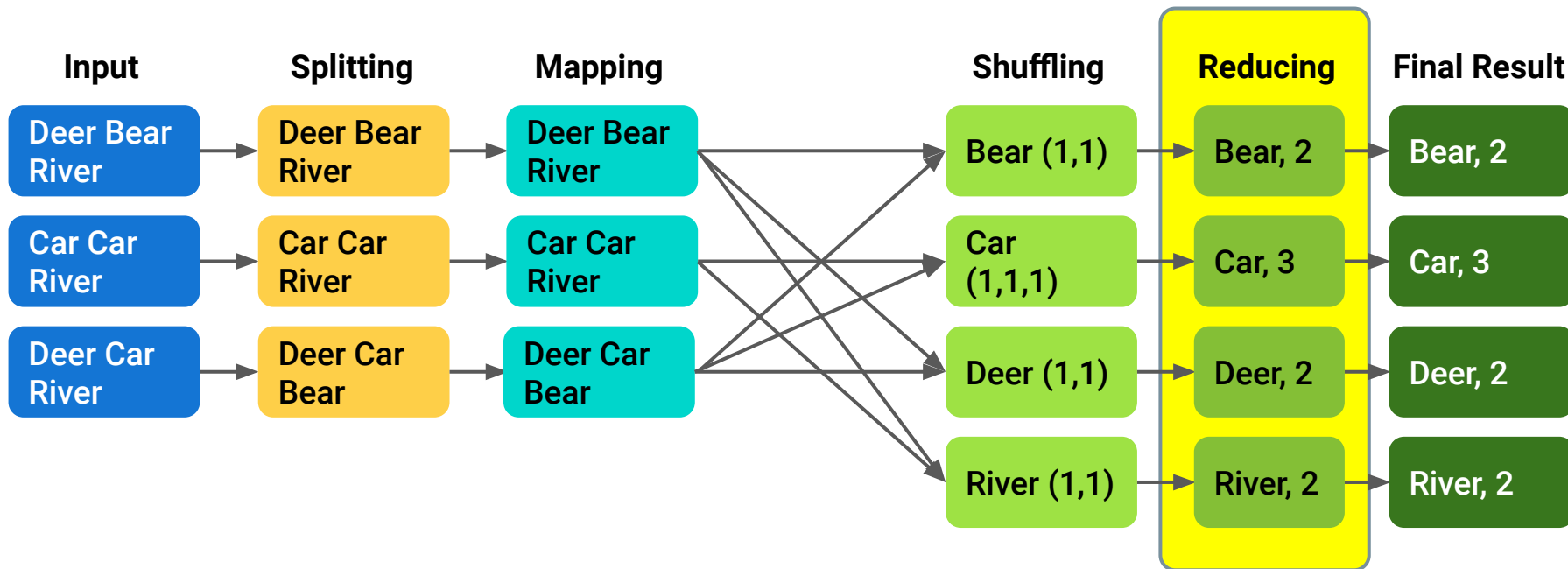
# MapReduce with MRJob (2 of 3)

**Mapping** means splitting up data, preprocessing it, and then converting it into key-value pairs.



# MapReduce with MRJob (3 of 3)

**Reducing** means aggregating the results.



# MapReduce: An Example Use Case

Imagine that our organization collects temperature data every day of the year for a city in the United States.

We can use MapReduce to find many aggregations.

But, one example is the average maximum temperature for each day over all the days in the dataset.

"2017-08-11"	97.5
"2017-08-12"	97.75
"2017-08-13"	98.5
"2017-08-14"	98.25
"2017-08-15"	97.25
"2017-08-16"	98.75
"2017-08-17"	99.0
"2017-08-18"	99.0
"2017-08-19"	99.0
"2017-08-20"	99.25
"2017-08-21"	97.25
"2017-08-22"	97.5
"2017-08-23"	98.75
"2017-08-24"	96.75
"2017-08-25"	91.25
"2017-08-26"	81.5
"2017-08-27"	74.25
"2017-08-28"	77.75
"2017-08-29"	86.0
"2017-08-30"	89.0
"2017-08-31"	91.25
"2017-09-01"	92.0
"2017-09-02"	92.5



# Questions?





## Activity: MapReduce Use Cases

In this activity, you'll find common use cases for MapReduce. The ability to research and learn about common use cases is an important skill to have in your career.

Suggested Time:

10 minutes



Time's Up! **Let's Review.**

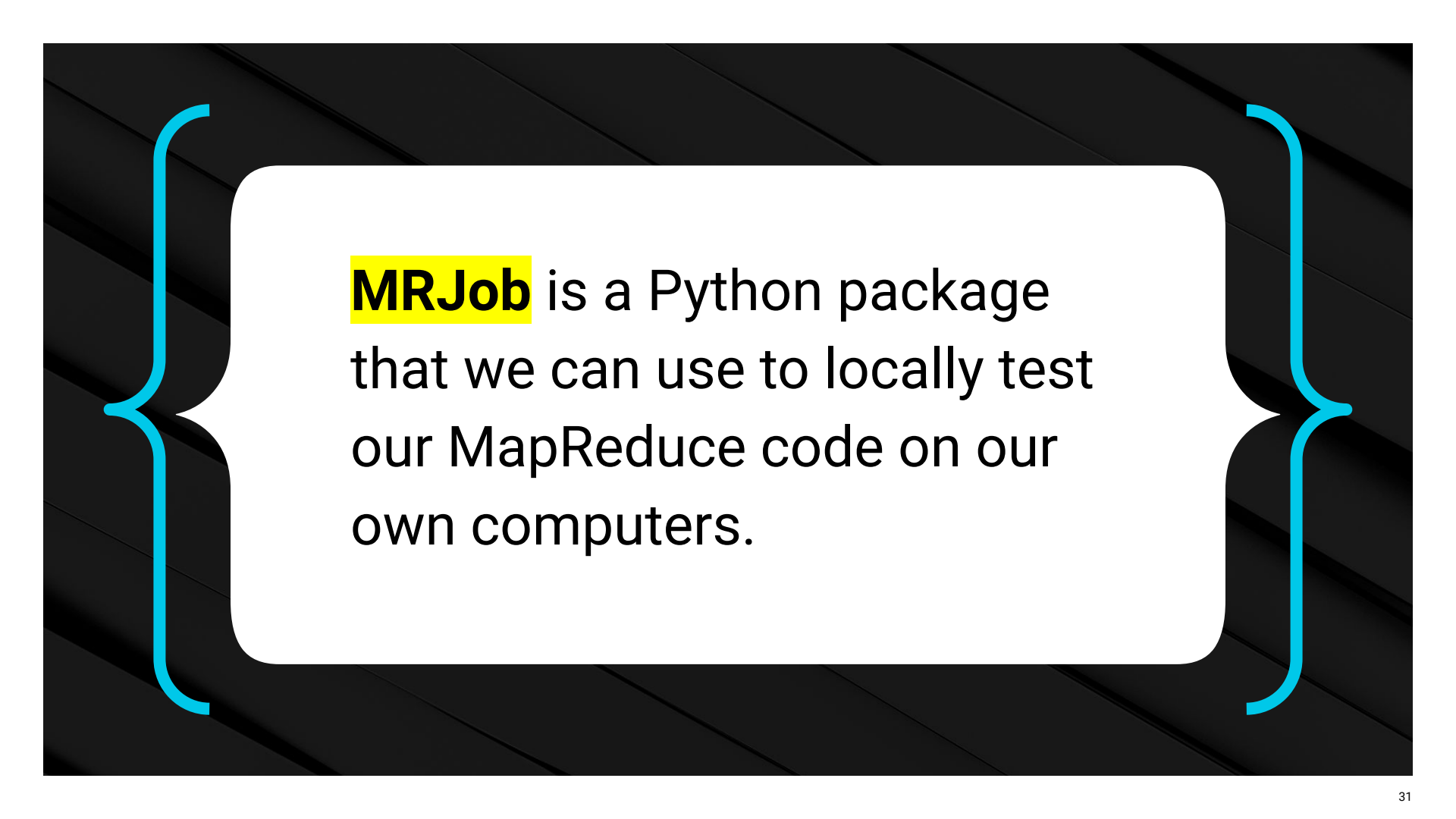
# Questions?



# MRJob with Python



**MapReduce is intended to distribute and process large data files.**



**MRJob** is a Python package  
that we can use to locally test  
our MapReduce code on our  
own computers.



## Group Programming Activity:

---

# MRJob with a CSV File

In this activity, you'll use MRJob with a CSV file to count the number of hot days in Austin, Texas.

Suggested Time:

15 Minutes



# Questions?





## Activity: Snow in Austin

In this activity, you'll use MRJob to return the days when it snowed in Austin, Texas.

Suggested Time:

15 minutes



Time's Up! Let's Review.

# Questions?





Break

# Spark Overview and Spark Installation Check



# Instructor Demonstration

---

## Spark Overview

# Spark: The Beginning

---

MapReduce was an amazing leap forward for handling massive amounts of data, but it was still slow.

So, the [AMPLab at UC Berkeley](#) came up with the idea of storing data in Resilient Distributed Datasets (RDDs) and using memory instead of disk space.

That improved the speed by 100 times.





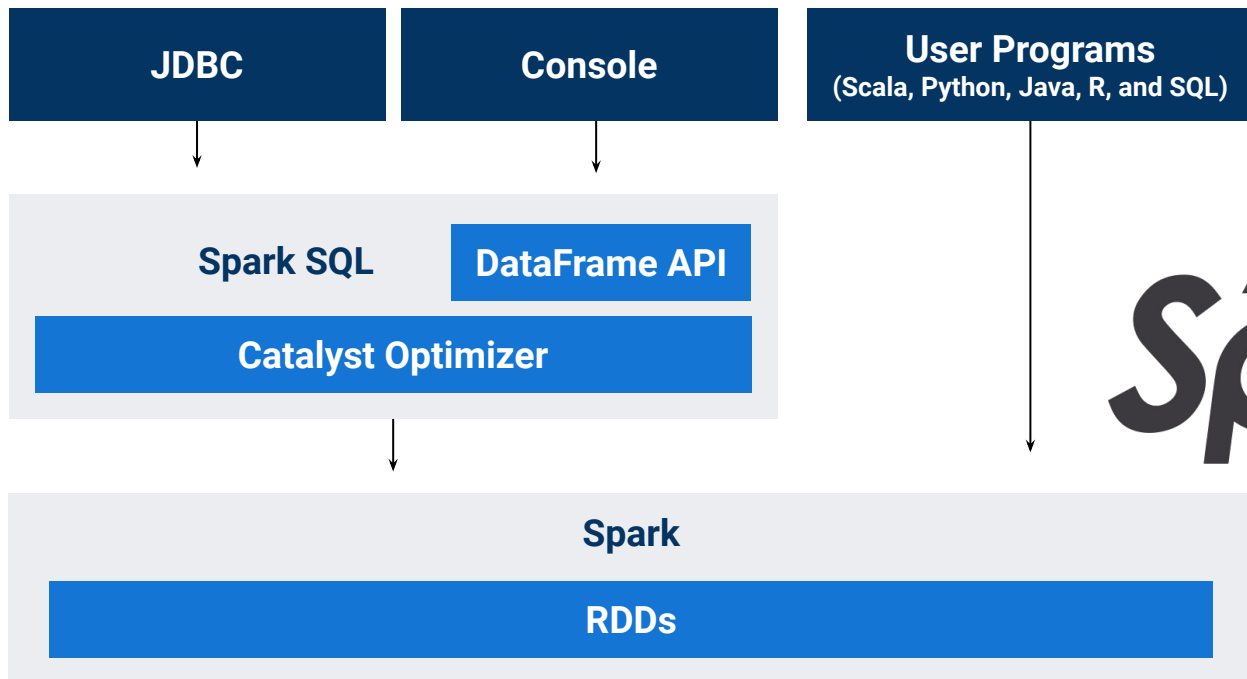
A stylized teal lightning bolt graphic with a 3D effect, pointing towards the text.

Although AMPLab developed Spark,  
they later donated it to the Apache Software  
Foundation—which now maintains it.

That's why you might find it called  
Apache Spark, just like you might find Hadoop  
called Apache Hadoop.

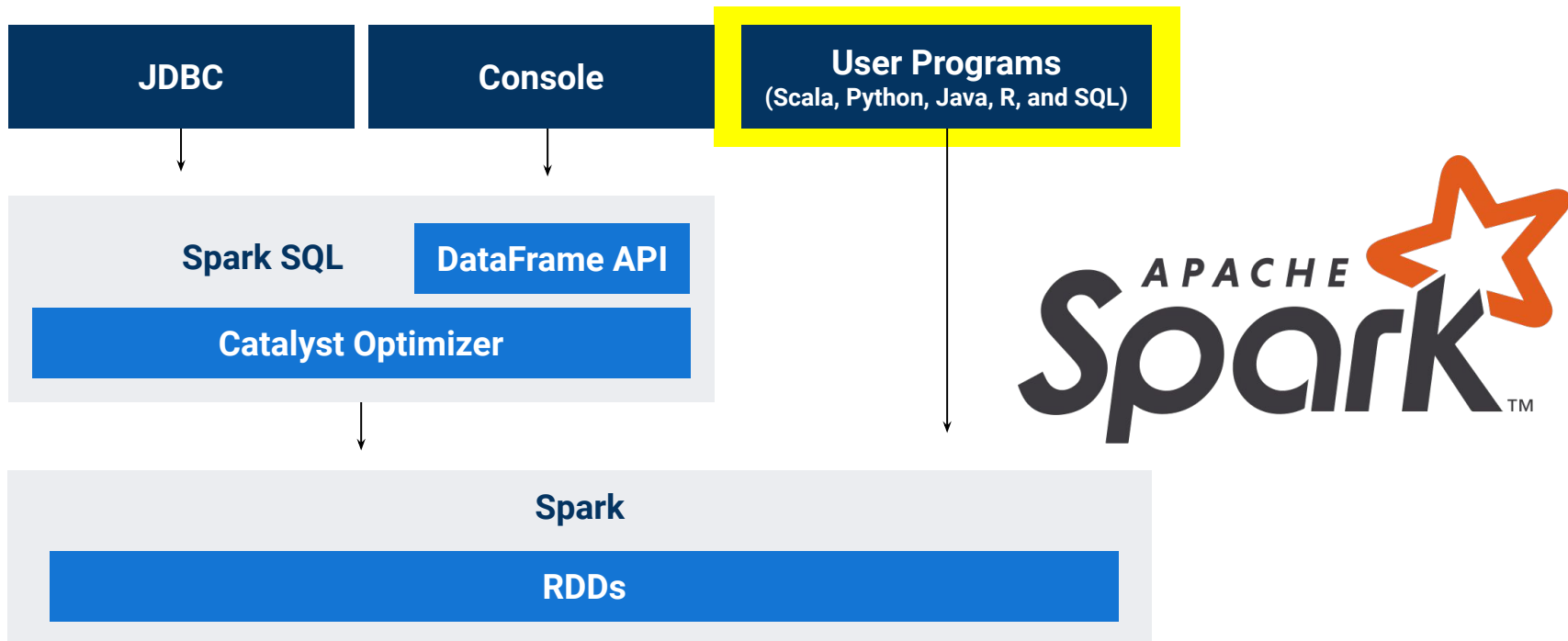
# Spark: The Evolution

As Spark evolved to its current version, 3.x, the direct use of RDDs was simplified by the use of datasets and DataFrames.



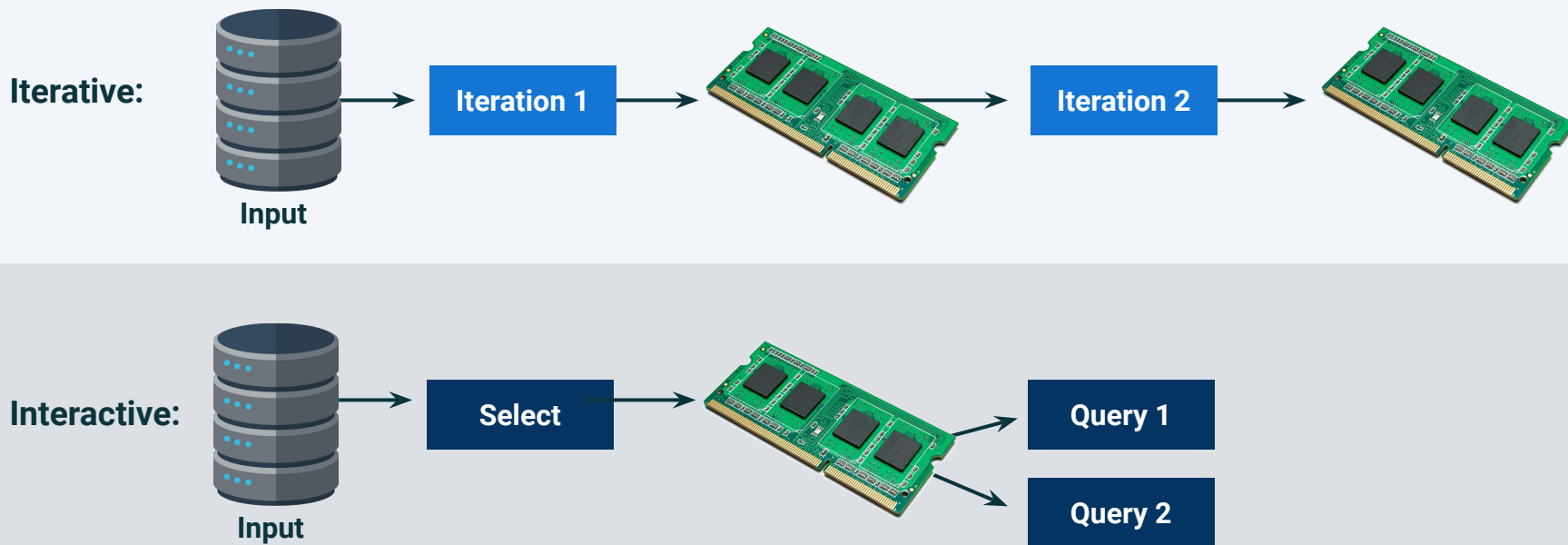
# Spark: Supported Languages and More

Spark can be programmed in Scala, Python, Java, R, and SQL. It has a rich ecosystem and is very scalable.



# Spark: In-Memory Computation

Spark uses in-memory computation instead of a disk-based solution. That means it has a limited need to read or write data from disks.



# Questions?





**Group Activity:**

---

# Spark Installation Check

In this activity, you'll check to make sure you have installed your dependencies to run PySpark in Jupyter notebook.

Suggested Time:

---

10 Minutes

# Questions?



# PySpark DataFrame Basics





# Instructor Demonstration

---

## PySpark DataFrame Basics

# Questions?





# Activity: Demographic DataFrame Basics

In this activity, you'll use PySpark DataFrame basics to analyze demographic data.

Suggested Time:

20 minutes



Time's Up! **Let's Review.**

# Questions?



# PySpark Filtering



# Instructor Demonstration

---

## PySpark Demographic Filtering

# Questions?







## Activity: PySpark Demographic Filtering

In this activity, you'll use the PySpark filtering functions to filter through the demographic dataset.

Suggested Time:

15 minutes



Time's Up! **Let's Review.**

# Questions?



*The  
End*