In Apache Spark, SparkContext and SparkSession are two important components used to establish connections with the Spark cluster and create a Spark application.

1. **SparkContext:**
   SparkContext is the entry point of a Spark application. It is the main component responsible for setting up the connection to the Spark cluster and coordinating the execution of tasks. In other words, it acts as a client to the Spark cluster manager, such as YARN or Mesos, and coordinates the distribution of the code across the cluster. SparkContext provides a programming interface for creating RDDs (Resilient Distributed Datasets) and executing operations on them.

2. **SparkSession:**
   SparkSession, on the other hand, was introduced in Spark 2.0 as a higher-level API that provides a unified entry point for working with Spark. It encompasses the functionality of SparkContext, Spark SQLContext, and HiveContext, making it easier to work with data in Spark. SparkSession provides a Spark execution environment that allows users to interact with Spark using SQL, DataFrames, and Datasets APIs.

The key differences between SparkContext and SparkSession are:

- SparkContext is the entry point for a Spark application, while SparkSession is a unified entry point for working with Spark.
- SparkContext is designed to work with RDDs, while SparkSession is designed to work with DataFrames and Datasets.
- SparkContext is a lower-level API that requires users to write more boilerplate code, while SparkSession provides a simpler, more convenient API for working with Spark.
- In Spark 2.x versions, SparkSession is a preferred way of creating Spark applications as it encapsulates SparkContext and other components.

To summarize, SparkContext is an essential component for a Spark application that provides a programming interface for working with RDDs, while SparkSession is a higher-level API that provides a unified entry point for working with Spark, enabling users to work with DataFrames and Datasets more easily.